



TA Document 2003004

AV/C Printer Subunit Specification 2.0

February 4 , 2004

Sponsored by:
1394 Trade Association

Accepted :
1394 Trade Association Board of Directors.

Abstract:
This document defines the AV/C Printer Subunit mode, specific data structures and command sets. The AV/C Printer Subunit may have not only the capability of receiving the still image or xHTML / HTML data via Asynchronous Serial Bus Connections, but also the capability of capturing the video stream via Isochronous connections. This document also defines the print job queuing mechanism.

This document describes printing a markup language data based on Print By Reference system model. The AV/C Printer Subunit may have the capability of retrieving the markup language data via Asynchronous Serial Bus Connections.

Keywords:
Video, Printer, Still image, xHTML, HTML, Print By Reference

1394 Trade Association Specifications are developed within Working Groups of the 1394 Trade Association, a non-profit industry association devoted to the promotion of and growth of the market for IEEE 1394-compliant products. Participants in working groups serve voluntarily and without compensation from the Trade Association. Most participants represent member organizations of the 1394 Trade Association. The specifications developed within the working groups represent a consensus of the expertise represented by the participants.

Use of a 1394 Trade Association Specification is wholly voluntary. The existence of a 1394 Trade Association Specification is not meant to imply that there are not other ways to produce, test, measure, purchase, market or provide other goods and services related to the scope of the 1394 Trade Association Specification. Furthermore, the viewpoint expressed at the time a specification is accepted and issued is subject to change brought about through developments in the state of the art and comments received from users of the specification. Users are cautioned to check to determine that they have the latest revision of any 1394 Trade Association Specification.

Comments for revision of 1394 Trade Association Specifications are welcome from any interested party, regardless of membership affiliation with the 1394 Trade Association. Suggestions for changes in documents should be in the form of a proposed change of text, together with appropriate supporting comments.

Interpretations: Occasionally, questions may arise about the meaning of specifications in relationship to specific applications. When the need for interpretations is brought to the attention of the 1394 Trade Association, the Association will initiate action to prepare appropriate responses.

Comments on specifications and requests for interpretations should be addressed to:

Editor, 1394 Trade Association
1111 South Main Street , Suite 100
Grapevine ,TX 76051, USA

1394 Trade Association Specifications are adopted by the 1394 Trade Association without regard to patents which may exist on articles, materials or processes or to other proprietary intellectual property which may exist within a specification. Adoption of a specification by the 1394 Trade Association does not assume any liability to any patent owner or any obligation whatsoever to those parties who rely on the specification documents. Readers of this document are advised to make an independent determination regarding the existence of intellectual property rights, which may be infringed by conformance to this specification.

Table of Contents

1. OVERVIEW.....	13
1.1 PURPOSE.....	13
1.2 SCOPE.....	13
2. REFERENCES.....	14
3. DEFINITIONS.....	15
3.1 CONFORMANCE LEVELS.....	15
3.2 GLOSSARY OF TERMS.....	15
3.3 ACRONYMS AND ABBREVIATIONS.....	17
3.4 NUMERICAL NOTATION.....	17
3.5 C++ CODE NOTATION.....	17
3.6 STATE MACHINE NOTATION.....	18
4. PROFILE.....	20
5. PRINTER SUBUNIT MODEL.....	21
5.1 PRINTER SUBUNIT LOGICAL MODEL.....	21
5.2 PRINTER SUBUNIT JOB QUEUING MODEL.....	21
5.3 PRINT BY REFERENCE MODEL.....	22
6. PRINTER SUBUNIT COMMANDS.....	24
6.1 PRINTER SUBUNIT TYPE.....	24
6.2 PRINTER SUBUNIT COMMANDS.....	24
6.2.1 <i>JOB QUEUE</i> command.....	25
6.2.2 <i>OPERATION MODE</i> command.....	43
6.2.3 <i>CAPTURE</i> command.....	55
6.2.4 <i>CAPTURE REF</i> command.....	82
6.2.5 <i>PRINTER STATUS</i> command.....	105
6.2.6 <i>VERSION</i> command.....	108
7. CONFIGURATION ROM FOR THE PRINTER SUBUNIT.....	114
7.1 CONFORMANCE.....	114
7.2 INSTANCE DIRECTORIES AND KEYWORDS.....	114
7.2.1 <i>Root directory</i>	114
7.2.2 <i>Instance directory</i>	115
7.2.3 <i>Keyword leaf</i>	115

List of Figures

Figure 3-1 – State machine notation	18
Figure 3-2 – Equivalent state machine	19
Figure 5-1 – Printer subunit logical model	21
Figure 5-2 – Printer subunit job queuing model	21
Figure 6-1 – Job queue model	26
Figure 6-2 – Print job state transitions	27
Figure 6-3 – JOB QUEUE CONTROL command	29
Figure 6-4 – JOB QUEUE STATUS command format	38
Figure 6-5 – JOB QUEUE STATUS response format	38
Figure 6-6 – OPERATION MODE CONTROL command	43
Figure 6-7 – OPERATION MODE STATUS command format	46
Figure 6-8 – OPERATION MODE STATUS response format	47
Figure 6-9 – CAPTURE CONTROL command basic format	56
Figure 6-10 – CAPTURE CONTROL command operation state machine	62
Figure 6-11 – CAPTURE STATUS command frame with subfunction get_status	73
Figure 6-12 – CAPTURE status response frame with subfunction get_status	74
Figure 6-13 – CAPTURE STATUS command frame with subfunction query	76
Figure 6-14 – CAPTURE status response frame with subfunction query	77
Figure 6.15 – CAPTURE REF CONTROL command basic format	82
Figure 6-16 – CAPTURE REF CONTROL command operation state machine	86
Figure 6-17 – CAPTURE REF STATUS command frame with subfunction get_status	99
Figure 6-18 – CAPTURE REF status response frame with subfunction get_status	100
Figure 6-19 – CAPTURE REF STATUS command frame with subfunction query	102
Figure 6-20 – CAPTURE REF status response frame with subfunction query	102
Figure 6-21 – PRINTER STATUS STATUS command frame	105
Figure 6-22 – PRINTER STATUS STATUS command response frame	105
Figure 6-23 – VERSION STATUS command frame	108
Figure 6-24 – VERSION STATUS command response frame	108
Figure 6-25 – subunit_version_information_dependent_field definitions	109
Figure 6-26 – subunit_version_information_dependent_field values	110
Figure 6-27 – VERSION STATUS command frame to get the latest version information	111
Figure 6-28 – VERSION STATUS command response frame to get the latest version information	111
Figure 6-29 – VERSION STATUS command frame to get the support level of the specified version	111
Figure 6-30 – VERSION STATUS command response frame to get the support level of the specified version	111
Figure 7-1 – Overall Structure of Instance Directories with Keyword leaves	114
Figure 7-2 – Instance directory	115
Figure 7-3 – Keyword leaf	115
Figure D.7-1 – Categorization of Image Data Format	120
Figure D.7-2 – Categorization of YCC 4:2:0/YCC 4:2:2, chunky/linear formats	121
Figure D.7-3 – Image Data Arrangement of sRGB raw	121
Figure D.7-4 – Image Data Arrangement of sRGB raw, quadlet aligned	122
Figure D.7-5 – Image Data Arrangement of YCC 4:2:2 raw, chunky	123
Figure D.7-6 – Image Data Arrangement of YCC 4:2:0 raw, chunky	123
Figure D.7-7 – Image Data Arrangement of YCC 4:2:2 raw, linear	124
Figure D.7-8 – Image Data Arrangement of YCC 4:2:0 raw, linear	125
Figure D.7-9 – Image Data Arrangement of Exif	126
Figure D.7-10 – Image Data Arrangement of JFIF	127
Figure D.7-11 – Image Data Arrangement of Exif	128
Figure D.7-12 – Image Data Arrangement of JPEG	129

List of Tables

Table 1.1 – Content change for version 1.0 R0.85	8
Table 1.2 – Content change for version 1.0 R0.90	8
Table 1.3 – Content change for version 1.0 R0.90	9
Table 1.4 – Content change for version 1.0 R0.96	9
Table 1.5 – Content change for version 1.0 R0.97	10
Table 1.6 – Content change for version 1.0 R1.0 AVWG Candidate.....	10
Table 1.7 – Content change for version 1.0 R1.0 TA Candidate.....	10
Table 1.8 – Content change for version 2.0 R0.00	10
Table 1.9 – Content change for version 2.0 R0.90	11
Table 1.10 – Content change for version 2.0 R0.91	11
Table 3.1 – Specific expression summary	18
Table 4.1 – profile name.....	20
Table 5.1 – Printer subunit implementation types	22
Table 6.1 – Support level of printer subunit commands.....	24
Table 6.2 – subfunction definitions and support level of “JOB QUEUE” command.....	29
Table 6.3 – status field definitions	30
Table 6.4 – result field definitions.....	30
Table 6.5 – print_job_ID definition.....	31
Table 6.6 – Successful command-response for the add_job subfunction	33
Table 6.7 – Failed command-response for the add_job subfunction.....	33
Table 6.8 – Possible result value for the rejected add_job subfunction.....	34
Table 6.9 – Successful command-response for the cancel_job subfunction.....	34
Table 6.10 – Failed command-response for the cancel_job subfunction.....	35
Table 6.11 – Possible result value for the rejected cancel_job subfunction	35
Table 6.12 – Successful command-response for the close_job subfunction.....	36
Table 6.13 – Failed command-response for the close_job subfunction.....	37
Table 6.14 – Possible result value for the rejected close_job subfunction	37
Table 6.15 – Successful command-response for STATUS command with specified print_job_ID.....	39
Table 6.16 – Failed command-response for STATUS command with specified print_job_ID.....	40
Table 6.17 – Possible result value for the rejected STATUS command with specified print_job_ID	40
Table 6.18 – Successful command-response for STATUS command with specified job_position	41
Table 6.19 – Failed command-response for the get_job subfunction with job_position	42
Table 6.20 – Possible result value for the rejected STATUS command	42
Table 6.21 – result field definitions.....	44
Table 6.22 – operation_mode_parameters definition	44
Table 6.23 – Successful command-response for the OPERATION MODE CONTROL command.....	45
Table 6.24 – Failed command-response for the OPERATION MODE CONTROL command.....	45
Table 6.25 – Possible result value for the rejected OPERATION MODE CONTROL command.....	46
Table 6.26 – subfunction field definitions.....	47
Table 6.27 – Successful command-response for the OPERATION MODE STATUS command with the subfunction of get_current_parameter.....	48
Table 6.28 – Failed command-response for the OPERATION MODE STATUS command with the subfunction of get_current_parameter.....	48
Table 6.29 – Possible result value for the rejected OPERATION MODE STATUS command with the subfunction of get_current_parameter.....	49
Table 6.30 – Successful command-response for the OPERATION MODE STATUS command with the subfunction of query_support_level	49
Table 6.31 – Possible result value for the STABLE response for OPERATION MODE STATUS command with the subfunction of query_support_level	49
Table 6.32 – Failed command-response for the OPERATION MODE STATUS command.....	50
Table 6.33 – Possible result value for the rejected OPERATION MODE STATUS command	50
Table 6.34 – Definitions of multiple_tiled	51
Table 6.35 – Definitions of speed_quality.....	52
Table 6.36 – Definitions of mono_color	53

Table 6.37 – Definitions of sizing	53
Table 6.38 – Definitions of orientations	54
Table 6.39 – Definitions of posx	54
Table 6.40 – Definitions of posy	55
Table 6.41 – Default value of operation_mode_parameters	55
Table 6.42 – subfunction field definitions for CAPTURE CONTROL command	56
Table 6.43 – status field definitions.....	57
Table 6.44 – result field definitions.....	58
Table 6.45 – data size field definitions	59
Table 6.46 – image_format_specifier definitions	60
Table 6.47 – Successful command-response for the CAPTURE CONTROL command with the subfunction of receive	65
Table 6.48 – Failed command-response for CAPTURE CONTROL command with the subfunction of receive.....	66
Table 6.49 – Possible result value for the rejected receive subfunction	67
Table 6.50 – Successful command-response for the CAPTURE CONTROL command with the subfunction of abort.....	68
Table 6.51 – Failed command-response for CAPTURE CONTROL command with the subfunction of abort.....	69
Table 6.52 – Possible result value for the rejected abort subfunction.....	70
Table 6.53 – Successful command-response for the CAPTURE CONTROL command with the subfunction of resume	71
Table 6.54 – Failed command-response for CAPTURE CONTROL command with the subfunction of resume.....	72
Table 6.55 – Possible result value for the rejected resume subfunction	72
Table 6.56 – subfunction field definitions for CAPTURE STATUS command.....	74
Table 6.57 – Definitions of other fields	75
Table 6.58 – Successful command-response for the CAPTURE STATUS command with the subfunction of get_status	78
Table 6.59 – status response field values.....	78
Table 6.60 – Failed command-response for the CAPTURE STATUS command with the subfunction of get_status	79
Table 6.61 – Possible result value for the rejected get_status subfunction.....	79
Table 6.62 – Successful command-response for the CAPTURE STATUS command with the subfunction of query	80
Table 6.63 – Failed command-response for the CAPTURE STATUS command with the subfunction of query	81
Table 6.64 – Possible result value for the rejected query subfunction.....	81
Table 6.65 – subfunction field definitions for CAPTURE REF CONTROL command.....	83
Table 6.66 – status field definitions.....	83
Table 6.67 – result field definitions.....	84
Table 6.68 – content_format_specifier definitions	85
Table 6.69 – Successful command-response for the CAPTURE REF CONTROL command with the subfunction of retrieve.....	90
Table 6.70 – Failed command-response for CAPTURE REF CONTROL command with the subfunction of retrieve.....	91
Table 6.71 – Possible result value for the rejected receive subfunction	92
Table 6.72 – Successful command-response for the CAPTURE REF CONTROL command with the subfunction of abort.....	93
Table 6.73 – Failed command-response for CAPTURE REF CONTROL command with the subfunction of abort.....	94
Table 6.74 – Possible result value for the rejected abort subfunction.....	95
Table 6.75 – Successful command-response for the CAPTURE REF CONTROL command with the subfunction of resume.....	96
Table 6.76 – Failed command-response for CAPTURE REF CONTROL command with the subfunction of resume.....	97

Table 6.77– Possible result value for the rejected resume subfunction	98
Table 6.78 – subfunction field definitions for CAPTURE REF STATUS command	100
Table 6.79 – Definitions of other fields.....	101
Table 6.80 – Successful command-response for the CAPTURE REF STATUS command with the subfunction of get_status.....	103
Table 6.81 – status response field values	103
Table 6.82 – Failed command-response for the CAPTURE REF STATUS command with the subfunction of get_status.....	104
Table 6.83 – Possible result value for the rejected get_status subfunction.....	104
Table 6.84 – status_category definitions	106
Table 6.85 – Recommended error messages	106
Table 6.86 – Successful command-response for the PRINTER STATUS STATUS command	107
Table 6.87 – Failed command-response for the PRINTER STATUS STATUS command	107
Table 6.88 – subunit_version_information field definitions	108
Table 6.89 – implementation_profile_id field definitions	109
Table 6.90 – Supported image formats and sizes	109
Table 6.91 – Supported markup language type	110
Table 6.92 – Successful command-response for the VERSION STATUS command to get the latest version information	112
Table 6.93 – Failed command-response for the VERSION STATUS command to get the latest version information	112
Table 6.94 – Successful command-response for the VERSION STATUS command to get the support level of specified version	112
Table 6.95 – Failed command-response for the VERSION STATUS command to get the support level of specified version.....	113
Table B.1 – Data transports for an AV/C printer subunit.....	117
Table C.1 – Basic sequence for one page print by asynchronous connection	118
Table C.2 – Enhanced sequence for index print.....	119
Table C.3 – Optional sequence for one page print using isochronous connection	119
Table D.4 – Additional Restrictions of Exif 2.1	126
Table D.5 – Exif Tag handling	126
Table D.6 – Additional Restrictions of JFIF	127
Table D.7 – Additional Restrictions of Exif 2.1	128
Table D.8 – Exif Tag handling.....	128
Table D.9 – Additional Restrictions of JPEG.....	129
Table H.1 – Basic sequence for markup data pull print profile by asynchronous connection.....	134

Change history

The following table shows the change history for this specification.

Version 1.0 R0.85(2000/02/28)

Table 1.1 – Content change for version 1.0 R0.85

Category	Description
Technical	Added the Scope, to reflect the copyright issue.
Technical	Updated the job queue state machine, Figure 5.2-2 & 5.2-3.
Technical	Changed the Table 5.2-3 to reflect the state machine change.
Technical	Changed the PRINTER STATUS command for status_category & status_information. Now the status_information is “status_category” dependent.
Technical	VERSION command opcode is decided to be 0xDF as subunit command range (General 3.0 missed to define the 0xE0~ED range?)
Technical	VERSION command is extended to get the support level of the specified version, in addition to the latest-version retrieval.
Technical	Added the CAPTURE command to support the extended_subunit_type, changed the arrangement of the operand[] for both control and STATUS command.
Technical	Added the subunit_type/subunit_ID/source_plug usage rule.

Version 1.0 R0.90 (2000/03/06)

Release for the Korea off-cycle meeting review

Table 1.2 – Content change for version 1.0 R0.90

Category	Description
Technical	Removed the subunit_type/subunit_ID/source_plug to reflect the opinions from WP members. Other mechanism to ensure the end-to-end operation will be proposed by the printer/camera_storage WP members. This topic (exclusive access for robustness) should be solved in other documents to come.
Technical	Description here
Technical	Added the job queue mechanism explanation.
Technical	Changed the definition of the PRINTER STATUS error_information field to message_number.
Editorial	Clarified the image format definitions and color space.
Editorial	Made the error code master index for the printer subunit commands.
Technical	Added the CAPTURE STATUS command response frame detail. (get_status & query)

Version 1.0 R0.95(2000/03/14)

Internal editorial checks by an editor.

Table 1.3 – Content change for version 1.0 R0.90

Category	Description
Editorial	Added the description of “opened job”. Timeout procedure is the same as “cancel”
Editorial	Deleted “any available plug” addressing from CAPTURE CONTROL command.
Editorial	Changed the alignments of “operation_mode_parameters”, correct the offset address.
Editorial	Changed the image_size_x and image_size_y definitions. These values SHALL be set to specify the image size even if it is not raw data.
Editorial	Clarify the usage of image_format_specifier “Unit plug defined” and “Don’t care” value usage.
Editorial	Changed the title, Annex: “Transformation...” → “Relationship between RGB values and YCC...”
Editorial	Changed the Image Width and Image Length definitions in the Table 5.2-68
Editorial	Annex D1. Overview: “The RGB color space is recommended to use of sRGB” → “The sRGB color space may be used as the RGB color space.”
Editorial	Added the descriptions for Connection setup procedure in Annex.
Editorial	Reassign the Table & Figure numbers.
Editorial	Correct other editorial errors

Version 1.0 R0.96 (2000/05/09)

Internal editorial checks by an editor.

Table 1.4 – Content change for version 1.0 R0.96

Category	Description
Editorial	Check the usage of “a” and “the”.
Editorial	Correct many editorial errors
Technical	Changed the job queue definitions. The job is queued when the job entry is created with add_job.
Technical	Extend operation mode parameters and its arrangements
Technical	Added the subfunction field for OPERATION MODE STATUS command to add query_support_level subfunction. Because the SPECIFIC INQUIRY command can NOT be used to get the support level of the parameters if the print_job_ID value is unspecified (if the print_job_ID = 0xFF...FF in the SPECIFIC INQUIRY CONTROL command, the target will return NOT IMPLEMENTED response like CONTROL command).

Version 1.0 R0.97 (2000/05/22)

Prepared for an additional work project meeting.

Table 1.5 – Content change for version 1.0 R0.97

Category	Description
Editorial	Correct many editorial errors
Editorial	Changed the character to the uppercase at the head of a sentence

Version 1.0 R1.0 – AVWG Candidate (2000/06/13)

Prepared for an AVWG Vote.

Table 1.6 – Content change for version 1.0 R1.0 AVWG Candidate

Category	Description
Editorial	Correct many editorial errors
Technical	Changed the opcode of VERSION to 0xB0 (Unit and Subunit command) to reflect the discussion at the offcycle AVWG at Sharp

Version 1.0 R1.0 TA Candidate (2000/07/26)

Reflected the results of AVWG Vote review.

Table 1.7 – Content change for version 1.0 R1.0 TA Candidate

Category	Description
Editorial	Correct many editorial errors
Technical	Changed the opcode of VERSION command format to allow the future extension to other subunits.

NOTE — The information on this page will be removed when this document is approved.

Version 2.0 R0.00(2003/01/13)

Table 1.8 – Content change for version 2.0 R0.00

Category	Description
Technical	VERSION command – Definition of profile that supports markup language data printing based on Print by Reference.
Technical	OPERATION MODE command – Definition of availability of XHTML / HTML Pull Print action
Technical	CAPTURE REF command – Addition of XHTML / HTML data formats
Technical	CAPTURE REF command – Definition of pull action
Technical	Definition of re-establishment procedure for Bus reset and exception process.

Version 2.0 R0.90(2003/04/28)

Table 1.9 – Content change for version 2.0 R0.90

Category	Description
Editorial	Add references 14 and 15 .
Editorial	Change the template to the latest one .
Editorial	Take back the field name " <i>number_of_pics</i> " to " <i>number_of_pics</i> "
Editorial	Repare the lacks of table frame
Editorial	Take back ""print " to "image " in Table of Difinitions of orientations
Editorial	Take back ""print " to "image " in Table of Difinitions of posx
Editorial	Take back ""print " to "image " in Table of Difinitions of posy
Technical	Delete " <i>data_size</i> " filed in CAPTURE REF command
Technical	Delete result values related to <i>data_size</i> in CAPTURE REF command
Editorial	Correct the word "HTML4.0 Traditional " to "HTML4.0 Transitional"
Editorial	Correct the word "HTML4.01 Traditional " to "HTML4.01 Transitional"
Editorial	Correct mistypes in Figure of CAPTURE REF CONTROL command operation state machine
Technical	Delete result value related to <i>data_size</i> for the rejected receive subfunction in CAPTURE REF command
Technical	DELETE query subfunction of CAPTURE REF STATUS command
Technical	Modify implementation_profile_id field definitions .
Editorial	Modify Supported markup language type table in section 6.

Version 2.0 R0.91(2003/05/15)

Table 1.10 – Content change for version 2.0 R0.91

Category	Description
Editorial	Modifiy Table 4.2 . Change table name and delete implementation_profile_ID row in the table.
Technical	Modify CAPTURE REF command field
Editorial	Delete description related to not existing field in CAPTURE REF command
Technical	Add description that CAPTURE REF CONTROL command shall be issued only one time per one job.
Editorial	Correct many other editorial errors
Technical	Change query subfunction of CAPTURE REF STATUS command and add description its action
Editorial	Add the description about the command sequence and action in case that camera storage subunit2.0 is used in data source unit in Annex G

Version 2.0 R0.91(2003/09/10)

Table 1.11 – Content change for TA Candidate 1.0

Category	Description
Editorial	Modify some editorial errors.

Table 1.12 – Content change for BoD Candidate 1.0

Category	Description
Editorial	Modify some editorial errors.

1. Overview

1.1 Purpose

The purpose of AV/C Printer Subunit is to provide the means to control or monitor a printer device on an IEEE 1394 network by AV/C commands.

1.2 Scope

This document defines the AV/C Printer Subunit modes, specific data structures and command sets. An AV/C Printer Subunit provides the capability of receiving the still image data via asynchronous serial bus connections, and optionally the capability of capturing the video stream via isochronous connections. This document also defines a mechanism for print job queuing.

Note that an AV/C printer compliant with this specification may print out the contents data as it received. It is a responsibility of a controller or a source device to behave not to constitute an infringement of contents' copyright.

This document describes printing a markup language data based on Print By Reference system model. The AV/C Printer Subunit may have the capability of retrieving the markup language data via Asynchronous Serial Bus Connections.

2. References

The following standards contain provisions, which through reference in this document constitute provisions of this standard. All the standards listed are normative references. Informative references are given in Annex A. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based on this standard are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below.

- [R1] IEEE Std 1394-1995, Standard for a High Performance Serial Bus.
- [R2] IEC 61883-1, Consumer audio/video equipment – Digital interface – Part 1: General.
- [R3] AV/C Digital Interface Command Set General Specification, Version 3.0. TA document number 1998003.
- [R4] AV/C Compatible Asynchronous Serial Bus Connections 2.0. TA document number 2000005
- [R5] AV/C Commands for Management of Enhanced Asynchronous Serial Bus Connections 1.0. TA document number 1999037
- [R6] AV/C Connection and Compatibility Management Specification 1.0. TA document number 1999031
- [R7] Configuration ROM for AV/C Devices 1.0, TA document number 1999027
- [R8] IEC 61966-2-1 (1999-10) - Multimedia systems and equipment - Colour measurement and management - Part 2-1: Colour management - Default RGB colour space – sRGB
- [R9] ITU-R BT.601-5 - Studio encoding parameters of digital television for standard 4:3 and wide-screen 16:9 aspect ratios
- [R10] ITU-R BT.709-4 - Parameter values for the HDTV standards for production and international programme exchange
- [R11] JEIDA-49-1998 Digital still camera image file format standard (exif)
- [R12] JPEG File Interchange Format, Version 1.02, C-Cube Microsystems
- [R13] ISO/IEC 10918-1 / ITU-T Recommendation T.81 information technology - Digital compression and coding of continuous-tone still images - Requirements and guide-lines
- [R14] IEEE-ISTO/PWG XHTML-Print Specification, Proposed Standard 5102.1 March 31, 2003
- [R15] IEEE-ISTO/PWG CSS Print Profile Specification, Proposed Standard 5102.2 March 31, 2003
- [R16] RFC 2396 August 1998, Uniform Resource Identifier (URI)

3. Definitions

3.1 Conformance Levels

3.1.1 expected: A key word used to describe the behavior of the hardware or software in the design models *assumed* by this Specification. Other hardware and software design models may also be implemented.

3.1.2 may: A key word that indicates flexibility of choice with *no implied preference*.

3.1.3 shall: A key word indicating a mandatory requirement. Designers are *required* to implement all such mandatory requirements.

3.1.4 should: A key word indicating flexibility of choice with a strongly preferred alternative. Equivalent to the phrase *is recommended*.

3.1.5 reserved fields: A set of bits within a data structure that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall zero these fields. Future revisions of this specification, however, may define their usage.

3.1.6 reserved values: A set of values for a field that are defined in this specification as reserved, and are not otherwise used. Implementations of this specification shall not generate these values for the field. Future revisions of this specification, however, may define their usage.

3.2 Glossary of Terms

3.2.1 Asynchronous: asyn “not same”– chronous – “time”. Asynchronous is an adjective used to describe data transfers are not sent at fixed time intervals. Asynchronous transfers are usually used for time insensitive data such as control commands.

3.2.2 Asynchronous connection: A logical point-to-point communication path established between producer and consumer nodes, that supports robust high-bandwidth, flow-controlled transfers of one or more data frames.

3.2.3 Asynchronous-connection consumer (abbreviated as consumer): The component of a node that consumes data frames provided by the asynchronous connection producer.

3.2.4 Asynchronous-connection producer (abbreviated as producer): The component of a node that produces data frames for consumption by the asynchronous connection consumer.

3.2.5 AV/C: Audio/video control. The AV/C Digital Interface Command Set of which a part is specified by this document.

3.2.6 AV/C Subunit: A part of an AV/C unit that is uniquely defined and offers a subset of functions that belong to the unit.

3.2.7 AV/C Unit: An electronic device that deals with Audio and/or Video data, *e.g.*, a camcorder or a VCR, attached as a Serial Bus node.

3.2.8 Byte: Eight bits of data, used as a synonym for octet.

3.2.9 Controller: A device at a serial bus node that sends AV/C commands to control a remote device.

3.2.10 CSR Architecture: A convenient abbreviation of the following reference (see clause 2): ISO/IEC 13213 : 1994 [ANSI/IEEE Std 1212, 1994 Edition], Information Technology—Microprocessor systems—Control and Status Register (CSR) Architecture for Microcomputer Buses.

3.2.11 Copy: The duplications of the page.

3.2.12 EUI-64: Extended Unique Identifier, 64-bits, as defined by the IEEE. The EUI-64 is a concatenation of the 24-bit company_ID obtained from the IEEE Registration Authority Committee (RAC) and a 40-bit number (typically a silicon serial number) that the vendor identified by company_ID guarantees to be unique for all of its products. The EUI-64 is also known as the node unique ID and is redundantly present in a node's configuration ROM in both the Bus_Info_Block and the Node_Unique_ID leaf.

3.2.13 Isochronous: iso – “same” chronous – “time”. Isochronous is an adjective used to describe data block transfers that occur at regular intervals. Isochronous transfers are used for time sensitive data such as audio and video.

3.2.14 Job: A basic entry for printing operation performed by a printer subunit. A job is created and controlled by a controller. A controller, which created a job successfully, is called as an owner of the job. The printing result of a job may be one or more pages.

3.2.15 Opened job: A job that had been created by JOB QUEUE CONTROL command with the subfunction of add_job, and had not been closed by JOB QUEUE CONTROL command with the subfunction of close_job.

3.2.16 Closed job: A job that had been closed by JOB QUEUE CONTROL command with the subfunction of close_job, and printing had not been finished yet.

3.2.17 Multiple tiled: Multiple print contents printed on a single page, but all of them shares the same print content data.

3.2.18 Nibble: Four bits of data. A byte is composed of two nibbles.

3.2.19 Node: An addressable device attached to Serial Bus with at least the minimum set of control registers defined by IEEE Std 1394–1995.

3.2.19 Node ID: A 16-bit number, unique within the context of an interconnected group of Serial Buses. The node ID is used to identify both the source and destination of Serial Bus asynchronous data packets. It can identify one single device within the addressable group of Serial Buses (unicast), or it can identify all devices (broadcast).

3.2.20 Orientation: Print direction for the page output. Some orientation value defined like “Landscape” or “Portrait” and so on.

3.2.21 Page: Basic output unit for the printer. Usually it means a sheet of paper. One or more print contents can be included in a single page.

3.2.22 Plug: A physical or virtual end-point connection implemented by an AV/C Unit or Subunit that may send or receive data. Plugs may be Serial Bus plugs, accessible through the Plug Control Registers (PCR's); they may be asynchronous plug; they may be external, physical plugs on the AV/C Unit; or they may be some virtual plugs implemented by the AV/C Subunits.

3.2.23 Queue: A first-in-first-out management mechanism for jobs. Once a job had been created, the job is put at the end of the queue inside a printer subunit. If the job becomes the first position, it is processed by a printer engine. Note that some printer implementation allows the queue which can queue only one job.

3.2.24 Quadlet: Four bytes of data.

3.2.25 Serial Bus: The hardware interconnects and software protocols for the peer-to-peer transport of serialized data, as defined by IEEE Std 1394–1995.

3.2.26 Target: A device at a serial bus node that receives and responds to AV/C commands from remote device.

3.2.27 Unit architecture: Software-visible resources that have a form and function, and describe a class of units. This document, in conjunction with the references above, defines the AV/C unit architecture for a class of AV/C devices.

3.2.28 PCR: An acronym of Plug Control Register.

3.2.29 Print By Reference:

3.3 Acronyms and Abbreviations

AV/C Audio Video Control

lsb least significant bit

LSB Least Significant Byte

msb most significant bit

MSB Most Significant Byte

3.4 Numerical notation

Decimal, hexadecimal, and binary numbers are used within this document. For clarity, decimal numbers are generally used to represent counts, hexadecimal numbers are used to represent addresses, and binary numbers are used to describe bit patterns within binary fields.

Decimal numbers are represented in their usual 0, 1, 2, ... format. Hexadecimal numbers are represented by a string of one or more hexadecimal (0-9,A-F) digits followed by the subscript 16, except in C++ code contexts, where they are written as `0x123EF2` etc. Binary numbers are represented by a string of one or more binary (0,1) digits, followed by the subscript 2. Thus the decimal number “26” may also be represented as “ $1A_{16}$ ” or “ 11010_2 ”.

3.5 C++ code notation

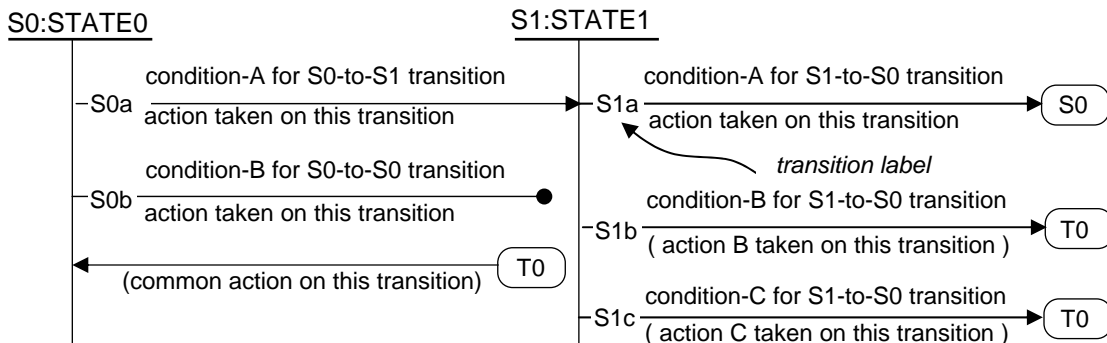
The conditions and actions of the state machines are formally defined by C++ code. Since many C++ code operators are non-obvious to the casual reader, their meanings are summarized in Table 3.1.

Table 3.1 – Specific expression summary

Expression	Description
$\sim I$	Bitwise complement of integer I
$++I$	Pre-increment of integer I (I is incremented, then used in the expression)
$--I$	Pre-decrement of integer I (I is decremented, then used in the expression)
$I \wedge J$	Bitwise XOR of integers I and J
$I \& J$	Bitwise AND of integer values I and J
$I J$	Bitwise OR of integer values I and J
$I \ll J$	Value of I, shifted left by J bits, zero fill
$I \gg J$	Value of I, shifted right by J bits, zero fill if I is an unsigned number, sign extension if I is signed
$I == J$	Equality test, true if I is equal to J
$I != J$	Inequality test, true if I is not equal to J
$!B$	Logical negation of boolean variable B
$A \&\& B$	Logical AND of boolean values A and B
$A B$	Logical OR of boolean values A and B

3.6 State machine notation

All state machines in this standard use the style shown in Figure 3-1. This is similar to the notation used in the Serial Bus standard, with modifications to more compactly and consistently illustrates state transition actions. To illustrative the functionality of transition-destination labels, the equivalent state machines without transition-destination labels is also illustrated in Figure 3-2. Labels of the form Sxx are state transition labels that signify a destination state. Labels of the form Tnn signify a transition to a tag which performs an action prior to entering the state to which it is attached.



Notes:
When appropriate, this note may specify that remaining state machine states are located and specified in other figures

Figure 3-1 – State machine notation

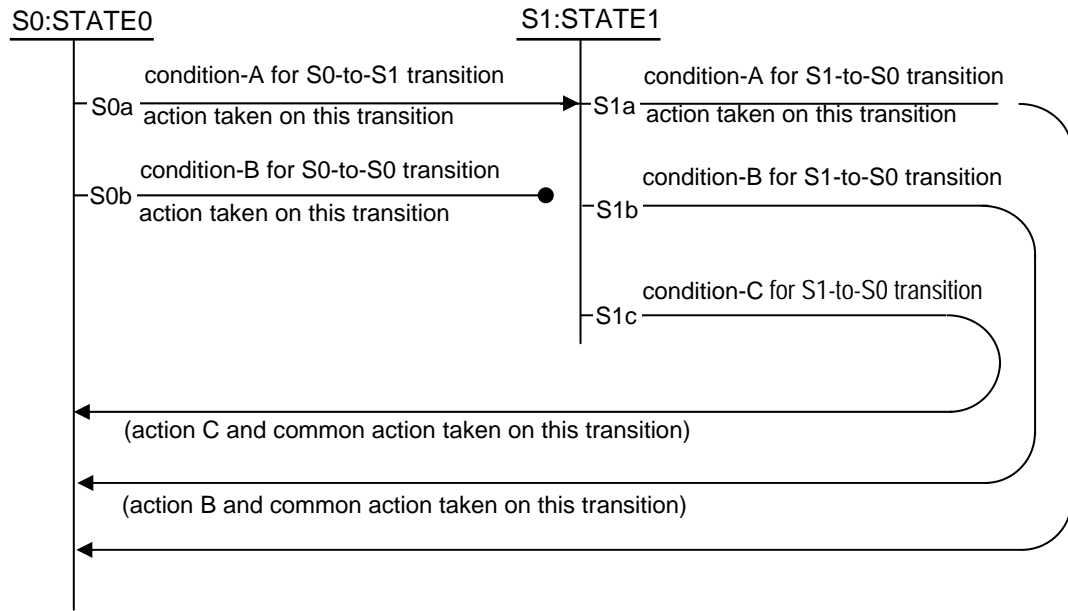


Figure 3-2 – Equivalent state machine

4. Profile

This specification introduces “profile” which defines not only supported print data format , e.g. the image formats and sizes or ,markup language type , but also support level of commands .

This specification defines two types of printer subunit profiles.

The profile names are shown in Table 4.1 below.

Table 4.1 – profile name

profile name	meaning
image push print	The subunit can print the image data based on traditional push print model.
markup data pull print	The subunit can print the markup language data based on Print By Reference model.

The printer subunit supporting the image push print profile can print the image data based on traditional push print model. The data shall be received via asynchronous inputs or may be received via isochronous stream depending on its implementation.

If the printer subunit supports only mandatory commands defined in the image push print profile, its functions are same as ones of printer subunit defined version 1.0 .

The printer subunit supporting the markup data pull print profile can print the markup language data based on Print By Reference model. The data shall be retrieved via asynchronous inputs .

The mandatory data format set is determined by the supported profile. For detailed information on support level, please refer to section 6.

The mandatory command set is determined by the supported profile. For detailed information on support level, please refer to section 6.

The supported profile is detected by the VERSION command described in section 6. The printer subunit may support only one profile or it may support multiple profiles.

5. Printer Subunit model

5.1 Printer subunit logical model

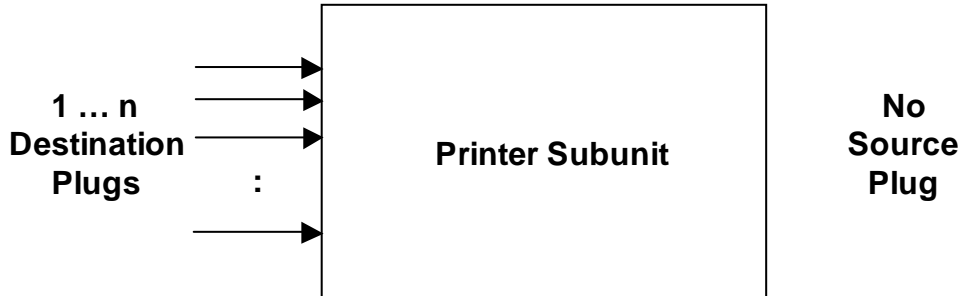


Figure 5-1 – Printer subunit logical model

A printer subunit has one or more destination plugs to receive print content to make hardcopies.

A printer subunit shall support asynchronous inputs, and it may support isochronous stream capture depending on its implementation.

A low-cost printer may support only an asynchronous input. On the other hand, a video printer may additionally support an isochronous video capture.

5.2 Printer subunit job queuing model

A printer subunit may have internal storage to store requested print jobs. An internal storage acts as a FIFO, which provides the job queuing capabilities.

Controllers that wish to print documents shall make a job using JOB QUEUE CONTROL command with the subfunction of add_job. Once a job is created, all of the print requests are addressed to the job entry until the job is closed using JOB QUEUE CONTROL command with the subfunction of close_job. The period between add_job and close_job is defined as “open” state.

After the job is closed by JOB QUEUE CONTROL command, it is automatically removed from the queue, when the print job is finished. The period between close_job and automatic removal is defined as “closed” state.

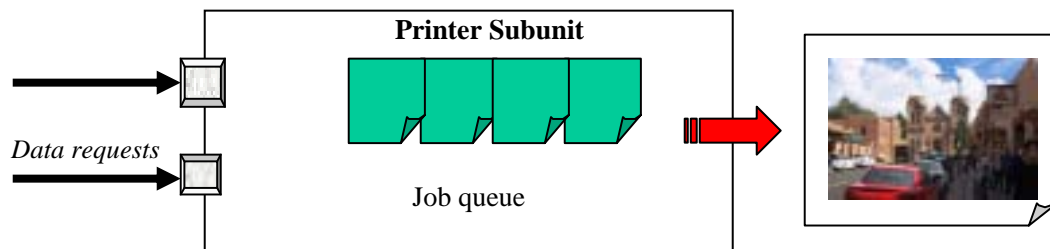


Figure 5-2 – Printer subunit job queuing model

If a printer subunit has capabilities to queue only one job, it is defined that the printer subunit has only one job queue entry. In this case, the printer subunit will not accept any additional job when it is processing a (single) job.

There are three possible implementations, as follows:

Table 5.1 – Printer subunit implementation types

Implementation type	Descriptions
Single job preparation, Single job queuing	This type of printer subunit can process only a single “opened” job, and it has no multi-job queuing capability. It prints received data without closing the job.
Single job preparation, Multi-job queuing (optional)	This type of printer subunit can handle only a single “opened” job, and it has queuing capabilities to queue more than one job.
Multi-job preparation, Multi-job queuing (optional)	This type of printer subunit can process multi “opened” jobs, and it has queuing capabilities to queue more than one job.

5.3 Print By Reference model

A following figure indicates the Print By Reference system model .

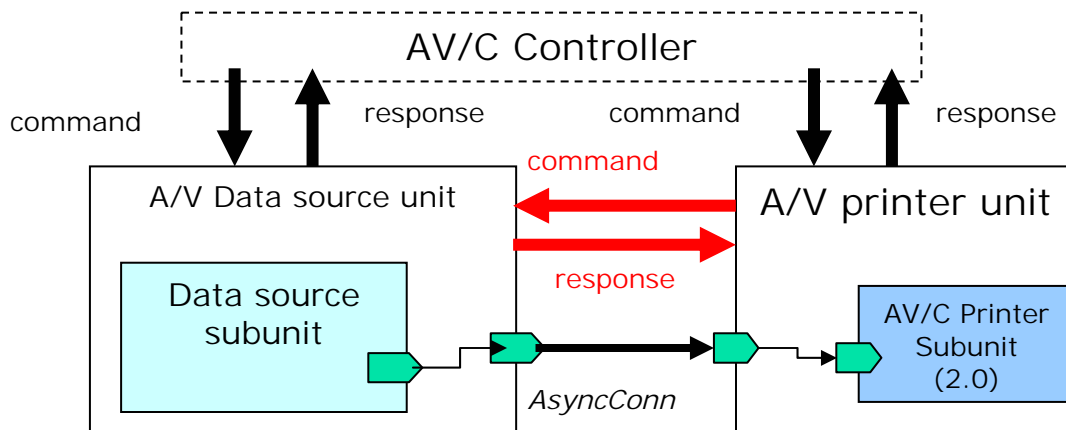


Figure 5.3 – Print By Reference logical system model

In the Print By Reference system model , job queuing model is same as defined in version 1.0 .

The difference between Print By Reference model and push image model defined in version 1.0 ,exists in process of capturing print data .

In push image model defined in version 1.0 , data source unit push image data to a printer subunit.

On the contrary, in Print By Reference model defined in version 2.0 , controller of data source unit requests a printer subunit to retrieve markup language data , with file information for specified the data.

Then the printer subunit acts as a controller for the data source subunit inside of data source unit ,and issues file retrieving command to the data source subunit , and retrieves the specified markup language data from the subunit.

The printer subunit may retrieve the markup language data at one time or may retrieve at several times .

The printer subunit may retrieve the data on convenient timing .

If the markup language data refers image data(s) inside, the printer subunit may retrieve the image data(s) in same way as markup language data.

In Print By Reference logical system model , two subunit , a printer subunit and a data source subunit ,e.g. a camera storage subunit , exists.

6. Printer subunit commands

6.1 Printer subunit type

The printer subunit shall be addressed by a subunit_type value of 02₁₆.

6.2 Printer subunit commands

This chapter defines printer subunit commands. The following Table 6.1 defines the support level of the printer subunit commands.

Table 6.1 – Support level of printer subunit commands

Opcode	Value	Support level (by ctype)						Comments
		image push print			markup data pull print			
		C	S	N	C	S	N	
JOB QUEUE	40 ₁₆	M ₁	M ₁	-	M ₁	M ₁	-	Create and manage the print job entries
OPERATION MODE	41 ₁₆	M ₁	M ₁	-	M ₁	M ₁	-	Set or get information about the operation mode of the printer subunit for the specified job
CAPTURE	42 ₁₆	M ₁	M ₁	-	O	O	-	Receive print image
PRINTER STATUS	43 ₁₆	-	M ₁	-	-	M ₁	-	Retrieve the printer operation status
CAPTURE REF	44 ₁₆	O	O	-	M ₁ ₋	M ₁ ₋	-	Retrieve markup language data by reference
VERSION	B0 ₁₆	-	M ₁	-	-	M ₁	-	Get or query the version information of the subunit

¹ Shall be supported by a printer subunit

A dash in the support level column indicates that the command is not defined for the *ctype* value CONTROL, STATUS or NOTIFY. The specific operand formats and corresponding response frame formats are described for each of the commands in the clauses that follow.

6.2.1 JOB QUEUE command

The JOB QUEUE command is used to create and manage a print job entries.

6.2.1.1 Overview of job and job queue managements

6.2.1.1.1 Job definition

When a controller wants one or more print contents to be printed out, it shall create a job by JOB QUEUE CONTROL command with the *subfunction* of `add_job`. All the requests to the printer subunit shall be addressed to the print job entry. After the controller creates the job entry, then it may set the operation modes of the printer (optional). Next, the controller inputs one or more image(s) via an asynchronous connections, an isochronous connections, external plug, or other subunit inside the AV unit. Or, the controller inputs one markup language data via an asynchronous connections, external plug, or other subunit inside the AV unit. After the controller had finished these operations, it requests the print job to be closed by JOB QUEUE CONTROL command with the *subfunction* of `close_job`. A period from a job creation (`add_job`) to a job closure (`close_job`) is defined as “opened”.

When the printer engine finished printing the job, the job disappears automatically. A period from job closure to job disappearance is defined as “closed”.

The job may be printed out after a job creation (concurrent with print content input(s)), or it may be printed out after a job closure. This specification defines a mechanism of job queue where one or more job can be queued in a printer subunit. A controller may request a job regardless of the job queuing capability of the printer subunit.

NOTE Multi job queuing capability is an optional feature. A printer subunit may or may not have the multi job queuing capability.

6.2.1.1.2 Job queue

As described above, a job (job entry) is not only defined as a basic entry for printing operation, but also defined as a basic entry for a queue management of the printer subunit.

Job queue is modeled as the FIFO (First-In-First-Out) queue. After a print job is created by the controller (JOB QUEUE CONTROL command with the *subfunction* of `add_job`), it is added at the end of the queue.

The following Figure 6-1 shows an example of the job queue. If three jobs already exist in the queue, their *job_position* values are assigned as 0002_{16} , 0001_{16} , and 0000_{16} . (The job which *job_position* value is zero, is being printed.)

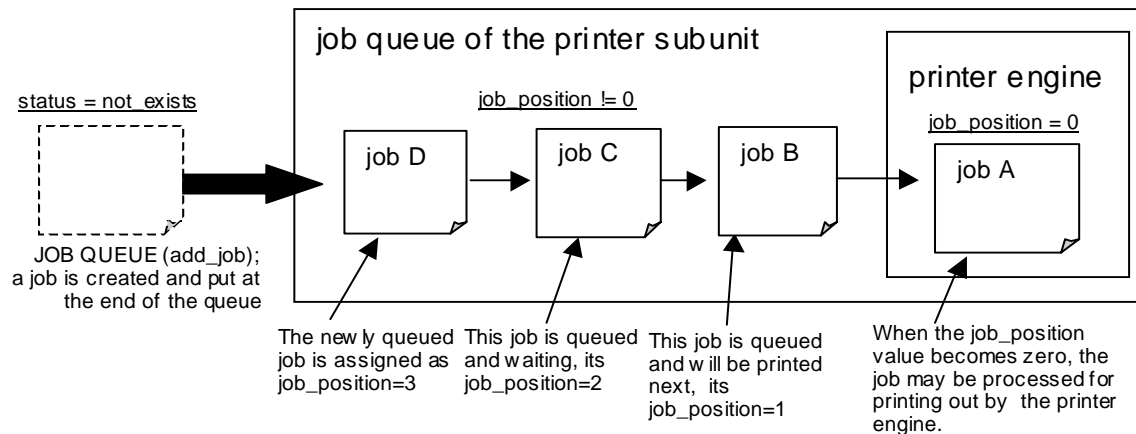


Figure 6-1 – Job queue model

When a job is newly created, it is positioned at the forth position and its `job_position` value is set to 0003_{16} . If the job with `job_position` value zero (the job being printed) has been printed out, then it will disappear from the queue, then all of the `job_position` numbers for each queued job are decremented by one. The job with `job_position` value zero (previously its `job_position` had been one), will be printed next. If a queued job is canceled by the `JOB QUEUE CONTROL` command with the *subfunction* of `cancel_job`, it will disappear from the queue, and the `job_position` values of other jobs having a larger value than the canceled job, will be decremented by one.

If there is no job exists in the queue, and if a job is created by `JOB QUEUE CONTROL` command with the *subfunction* of `add_job`, its `job_position` value is assigned to be zero. The printer subunit may start printing process when a first image had been received. For the markup language data , the printer subunit may start printing process when the markup language data and images referred in it had been retrieved as enough to print. For the markup language data , it may be not needed for starting of printing subunit to retrieve whole of print contents .

A job queue may store more than zero print jobs inside it. The capacity of the job queue may be differ by the printer's implementation. The capacity may be determined by the number of jobs in the queue, or by the total size of the queued job data size.

6.2.1.1.3 Job state transitions

The following Figure 6-2 shows the state transitions of a print job. Basically, the state transitions are affected by successful `JOB QUEUE CONTROL` command.

- 1) First, a controller makes a job entry by `JOB QUEUE CONTROL` command with the *subfunction* of `add_job`. The job is created in the queue and transitions to the **S1:opened_idle** state. If there is no job in the queue, the `job_position` value is zero, otherwise the `job_position` value is assigned to be at the end of the queue (`job_position` value is assigned to be the largest value).
- 2) While the job is in the **S1:opened_idle** state, the `OPERATION MODE CONTROL` command can be accepted to set up the print operation mode for the job.
- 3) After receiving the first `CAPTURE CONTROL` command with the *subfunction* of `receive`, the job transitions to the **S2:opened_capturing** state. When the job is in the **S2:opened_capturing** state, the one of more `CAPTURE CONTROL` command with the *subfunction* of `receive` can be accepted.

- 4) After the controller had finished capturing images, it issues the JOB QUEUE CONTROL command with the *subfunction* of close_job to finish the job preparing period. (**S3:closed**)
- 5) The job automatically disappears (transitions to the initial **S0:not_exists** state) when its printing process is completed.

Note that the job can be canceled by JOB QUEUE CONTROL command with the *subfunction* of cancel_job, whenever the CONTROL command is accepted, then the job transitions to the **S0:not_exists** state.

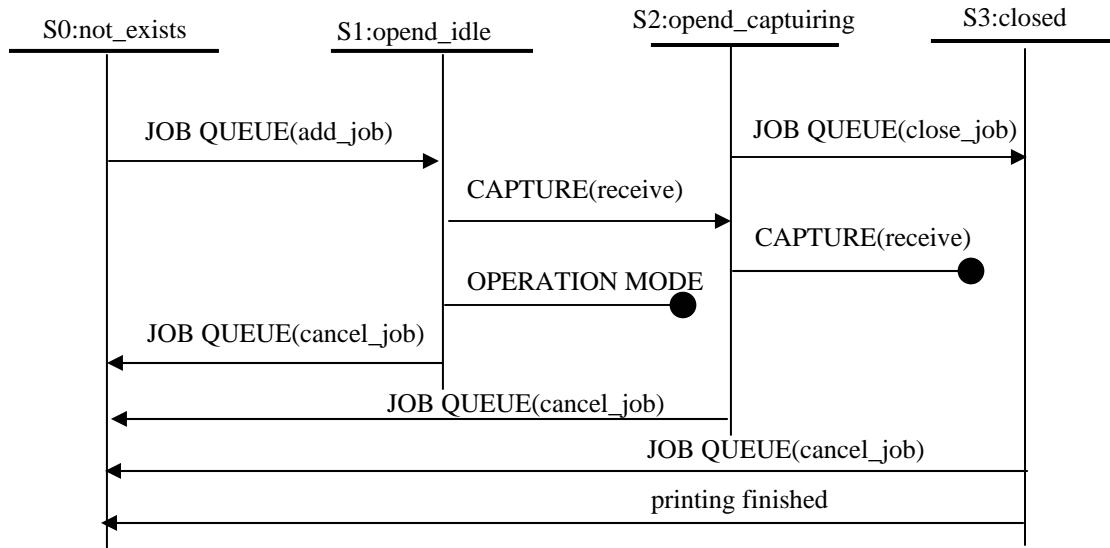


Figure 6-2 – Print job state transitions

NOTE A serial bus reset does not affect the state of jobs, because the job has the controller's EUI-64 value in the *print_job_ID* field. A controller can find out its job with JOB QUEUE STATUS commands. If the target had been executing the CAPTURE CONTROL command, the controller should restore the related connection first, then restore CAPTURE CONTROL command execution.

Basically, the state transitions affected by JOB QUEUE CONTROL command and CAPTURE REF CONTROL command , is same as one affected by JOB QUEUE CONTROL command and CAPTURE CONTROL command .

In the the state transitions , "CAPTURE REF CONTROL command with the *subfunction* of retrieve" only replace "CAPTURE CONTROL command with the *subfunction* of receive ".

6.2.1.1.4 Print job idle timeout

A controller should close a job by JOB QUEUE CONTROL command with the *subfunction* of close_job after it finished all capturing process. When a printer subunit had not received any CONTROL command or STATUS command addressed to the opened job (S1 or S2) for **30 seconds**, it may detect the job idle timeout and may cancel the job by itself to avoid hung-up.

In order to avoid above situation, a controller is recommended to perform as described below:

- 1) A controller should not generate an idle period between the command execution larger than 30 seconds.
- 2) The controller issues a STATUS command (e.g. JOB QUEUE STATUS command) to reset the timer of the specified job.

NOTE When a job is automatically canceled, a target may request the data transport layer to break the unused unit-to-unit connection or subunit-to-subunit connection, to release the connection related resource (e.g. unit plug). For example, if the single job type printer subunit detects the idle timeout, the printer subunit may break the asynchronous connection which had been already established to release the plug resource.

6.2.1.2 JOB QUEUE CONTROL command

The JOB QUEUE CONTROL command (ctype = CONTROL) is used to create, cancel, or close a job. The format of JOB QUEUE CONTROL command is illustrated by the figure below:

	length	msb						lsb
opcode	1	JOB QUEUE (40 ₁₆)						
operand[0]	1	subfunction						
operand[1]	1	status						
operand[2]	1	result						
operand[3]	2	Job_position						
operand[4]								
operand[5]	12	print_job_ID						
:								
operand[16]								
operand[17]	2	current_copies						
operand[18]								
operand[19]	2	requested_copies						
operand[20]								
operand[21]	2	current_pages						
operand[22]								
operand[23]	2	requested_pages						
operand[24]								
operand[25]	2	number_of_jobs						
operand[26]								

Figure 6-3 – JOB QUEUE CONTROL command

6.2.1.2.1 Field definitions

subfunction: This field indicates the action to be taken by the target. The detailed definition of this field is described in Table 6.2 as follows.

Table 6.2 – subfunction definitions and support level of “JOB QUEUE” command

Value	Symbol	Meaning
01 ₁₆	Add_job	Add a new job, specifying a <i>print_job_ID</i>
02 ₁₆	cancel_job	Cancel a job specified by <i>print_job_ID</i>
03 ₁₆	Close_job	Close a job, specifying a <i>print_job_ID</i>
Other values	-	Reserved

status: This field shall be FF₁₆ in the CONTROL command frame. In case of the response frame of a CONTROL command, this field indicates the status of the print job specified by the *print_job_ID* field. The detailed definition of this field is described in Table 6.3 as follows.

Table 6.3 – status field definitions

Value	Symbol	Meaning
00 ₁₆	not_exists	The specified job does not exist
01 ₁₆	opened_idle	The specified job is now just created. Ready to accept the operation mode change or print content capturing.
02 ₁₆	opened_capturing	The specified job is now in the capturing state.
03 ₁₆	closed	The specified job is now queued. waiting to be printed or being printed.
FF ₁₆	-	No meaning (used by command frame)
Other values	-	Reserved

result: This field shall be FF₁₆ in the CONTROL command frame. In case of the response frame of a CONTROL command, this field indicates the execution result of the CONTROL command. The detailed definition of this field is described in Table 6.4 as follows.

Table 6.4 – result field definitions

Value	Symbol	Meaning
00 ₁₆	no_error	The command execution had been finished successfully, or no command is under execution
86 ₁₆	no_job_exists	The print job with specified <i>print_job_ID</i> or <i>job_position</i> does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
92 ₁₆	queue_full	The command is rejected because the job queue is full.
93 ₁₆	job_already_exists	The add_job <i>subfunction</i> is rejected because the specified job already exists.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred
FF ₁₆	-	No meaning (used by command frame)
Other values	-	Reserved

job_position: This field shall be FFFF₁₆ in the CONTROL command frame. In case of the response frame of a CONTROL command, this field indicates the position of the job in the queue.

If the specified job is at the first entry of the job queue (this means the job may be processed for printing), the value of this field is zero.

The valid value range of this field is from 0000₁₆ to FFFE₁₆. The value of FFFF₁₆ indicates “No meaning”.

Target shall assign the *job_position* number one by one to each job being in the job queue.

The smaller *job_position* number shall be, the earlier the job shall be printed out.

When a job is newly created, it is positioned at the forth position. If three jobs are in the job queue (*job_position* = 0,1,2), its *job_position* value is set to 3.

If the job with *job_position* value zero (the job under printing) had been printed out, then it will disappear from the queue, then all of the *job_position* numbers for each queued job shall be decremented by one.

If a queued job is canceled, it will disappear from the queue, and the *job_position* values of other jobs, which had been bigger than the canceled job, shall be decremented by one.

print_job_ID: This field specifies an identifier of the print job requested by the controller or the subunit itself, defined in Table 6.5 as follows.

Table 6.5 – print_job_ID definition

Address Offset	Contents
00 ₁₆	owner_EUI64
01 ₁₆	
02 ₁₆	
03 ₁₆	
04 ₁₆	
05 ₁₆	
06 ₁₆	
07 ₁₆	
08 ₁₆	record_ID
09 ₁₆	
0A ₁₆	
0B ₁₆	

The controller is expected to set its EUI-64 value for *owner_EUI64* field to add the job, or to access the job entry. If the printer subunit had made the job by itself, *owner_EUI64* shall be zero.

The controller shall assign *record_ID* value for its internal job management. The value of *record_ID* field is expected to be assigned uniquely by the controller (from zero to 7FFFFFFF₁₆). The of msb of the *record_ID* field (80000000₁₆) is reserved for the command frame.

If a print job already exists in the queue, with the same *print_job_ID* value as specified by *add_job* subfunction, the *add_job* subfunction request is rejected with the result value of *job_already_exists*.

If the printer subunit received the JOB QUEUE CONTROL command which subfunction value is NOT *add_job*, and there is no job which has specified *print_job_ID* value, the printer subunit returns the REJECTED response for the CONTROL command with the result value of *no_job_exists*.

The value of FFFFFFFFFFFFFFFFFFFFFFFF₁₆ has no meaning.

current_copies: This field shall be FFFF₁₆ in the CONTROL command frame. In case of the response frame of a CONTROL command, this field indicates the sequence number of copies for a page specified by the *current_pages* and *print_job_ID*, which is currently being printed by the printer engine.

The value of FFFF₁₆ has no meaning and reserved for the command frame.

If the specified job is not being printed, this field value is set to zero.

If the first copy is being printed, this field value is set to one.

requested_copies: This field shall be FFFF₁₆ in the CONTROL command frame. In case of the response frame of a CONTROL command, this field indicates the requested number of copies for the job specified

by the *print_job_ID*, and this value is equal to the *number_of_copies* value defined in the “6.2.2.3.5 Number_of_copies”

The value of $FFFF_{16}$ has no meaning and reserved for the command frame.

current_pages: This field shall be $FFFF_{16}$ in the CONTROL command frame. In case of the response frame of a CONTROL command, this field is optionally returned and shall have value of $FFFF_{16}$ when not supported. When supported this field indicates the sequence number of pages currently being printed by the printer engine.

If the specified job is not being printed, this field value is set to zero.

requested_pages: This field shall be $FFFF_{16}$ in the CONTROL command frame. In case of the response frame of a CONTROL command, this field is optionally returned and shall have value of $FFFF_{16}$ when not supported. When supported this field indicates the requested number of pages in one copy for the job for the response frame. This value can be calculated as below:

```
ROUNDUP( "number of accepted CAPTURE CONTROL commands" / number_of_pic );
```

NOTE A *number_of_pict* value is specified by the *operation_mode_parameters* of the OPERATION MODE CONTROL command frame

If the specified job is not being printed, this field value may be set to zero.

number_of_jobs: This field shall be $FFFF_{16}$ in the CONTROL command frame. In case of the response frame of a CONTROL command, this field indicates the total number of jobs currently in the queue.

If there is no jobs in the queue, the value of this field is zero. If there is only one job at the first entry of the job queue and there is no other jobs in the queue, the value of this field is one.

The valid value range of this field is from 0000_{16} to $FFFE_{16}$. The value of $FFFF_{16}$ indicates “No meaning”.

6.2.1.2.2 JOB QUEUE CONTROL command operations

The following clauses defines the detail of CONTROL command and response. A controller shall issue the JOB QUEUE CONTROL commands as described below.

6.2.1.2.2.1 Add_job subfunction operation

An add_job subfunction is used to create a new job entry for the printer subunit manipulation. The controller shall add a new job prior to any operation to the printer subunit.

The following Table 6.6 shows the sample command-response for a successful add_job subfunction.

Table 6.6 – Successful command-response for the add_job subfunction

	opcode, operands	CONTROL command	ACCEPTED response
Opcode	JOB QUEUE	40 ₁₆	←
operand[0]	subfunction	01 ₁₆ (add_job)	←
operand[1]	status	FF ₁₆	01 ₁₆ (opened_idle)
operand[2]	result	FF ₁₆	00 ₁₆ (no_error)
operand[3]	Job_position	FFFF ₁₆	<i>current job position</i>
operand[4]			
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	current_copies	FFFF ₁₆	0000 ₁₆
operand[18]	requested_copies	FFFF ₁₆	0000 ₁₆
operand[19]			
operand[20]	current_pages	FFFF ₁₆	0000 ₁₆
operand[21]			
operand[22]	requested_pages	FFFF ₁₆	0000 ₁₆
operand[23]			
operand[24]	number_of_jobs	FFFF ₁₆	<i>current number of jobs</i>
operand[25]			
operand[26]			

The following Table 6.7 shows the sample command-response for a failed add_job subfunction.

Table 6.7 – Failed command-response for the add_job subfunction

	opcode, operands	CONTROL command	REJECTED response
opcode	JOB QUEUE	40 ₁₆	←
operand[0]	subfunction	01 ₁₆ (add_job)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	job_position	FFFF ₁₆	←
operand[4]			
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	current_copies	FFFF ₁₆	←
operand[18]	requested_copies	FFFF ₁₆	←
operand[19]			
operand[20]	current_pages	FFFF ₁₆	←
operand[21]			
operand[22]	requested_pages	FFFF ₁₆	←
operand[23]			
operand[24]	number_of_jobs	FFFF ₁₆	←
operand[25]			
operand[26]			

For the failed add_job subfunction, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.8 – Possible result value for the rejected add_job subfunction

Value	Symbol	Meaning
87 ₁₆	Busy	The command is rejected because the subunit is busy.
92 ₁₆	queue_full	The command is rejected because the job queue is full.
93 ₁₆	job_already_exists	The add_job subfunction is rejected because the specified job already exists.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.

6.2.1.2.2.2 Cancel_job subfunction operation

A cancel_job subfunction is used to cancel an existing job. The controller may cancel a job to abort the printing operations.

The following Table 6.9 shows the sample command-response for a successful cancel_job subfunction.

Table 6.9 – Successful command-response for the cancel_job subfunction

	opcode, operands	CONTROL command	ACCEPTED response
Opcode	JOB QUEUE	40 ₁₆	←
operand[0]	subfunction	02 ₁₆ (cancel_job)	←
operand[1]	Status	FF ₁₆	00 ₁₆ (not_exists)
operand[2]	Result	FF ₁₆	00 ₁₆ (no_error)
operand[3]	job_position	FFFF ₁₆	←
operand[4]			
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	current_copies	FFFF ₁₆	<i>current copies</i>
operand[18]			
operand[19]	requested_copies	FFFF ₁₆	<i>requested copies</i>
operand[20]			
operand[21]	current_pages	FFFF ₁₆	<i>current pages</i>
operand[22]			
operand[23]	requested_pages	FFFF ₁₆	<i>requested pages</i>
operand[24]			
operand[25]	number_of_jobs	FFFF ₁₆	<i>current number of jobs</i>
operand[26]			

The following Table 6.10 shows the sample command-response for a failed cancel_job subfunction.

Table 6.10 – Failed command-response for the cancel_job subfunction

	Opcode, operands	CONTROL command	REJECTED response
opcode	JOB QUEUE	40 ₁₆	←
operand[0]	subfunction	02 ₁₆ (cancel_job)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	job_position	FFFF ₁₆	←
operand[4]			
operand[5]	Print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	current_copies	FFFF ₁₆	←
operand[18]			
operand[19]	requested_copies	FFFF ₁₆	←
operand[20]			
operand[21]	current_pages	FFFF ₁₆	←
operand[22]			
operand[23]	requested_pages	FFFF ₁₆	←
operand[24]			
operand[25]	number_of_jobs	FFFF ₁₆	←
operand[26]			

For the failed cancel_job subfunction, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.11 – Possible result value for the rejected cancel_job subfunction

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified <i>print_job_ID</i> does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.

6.2.1.2.2.3 Close_job subfunction operation

A close_job subfunction is used to close a job in an opened state. The controller may close a job when there is no undergoing print content input process. If the target is executing the print content input process, the close_job subfunction will be rejected by REJECTED response with the result field value of busy.

The following Table 6.12 shows the sample command-response for a successful close_job subfunction.

Table 6.12 – Successful command-response for the close_job subfunction

	opcode, operands	CONTROL command	ACCEPTED response
Opcode	JOB QUEUE	40 ₁₆	←
Operand[0]	subfunction	03 ₁₆ (close_job)	←
Operand[1]	status	FF ₁₆	<i>current job status</i>
Operand[2]	result	FF ₁₆	00 ₁₆ (no_error)
Operand[3]	job_position	FFFF ₁₆	<i>current job position</i>
Operand[4]			
Operand[5]	Print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	current_copies	FFFF ₁₆	<i>current copies</i>
operand[18]	requested_copies	FFFF ₁₆	<i>requested copies</i>
operand[19]			
operand[20]	current_pages	FFFF ₁₆	<i>current pages</i>
operand[21]			
operand[22]	requested_pages	FFFF ₁₆	<i>requested pages</i>
operand[23]			
operand[24]	number_of_jobs	FFFF ₁₆	<i>current number of jobs</i>
operand[25]			
operand[26]			

NOTE “current job status” may be 00₁₆ (not_exists) or 03₁₆ (closed).

The following Table 6.13 shows the sample command-response for a failed close_job subfunction.

Table 6.13 – Failed command-response for the close_job subfunction

	Opcode, operands	CONTROL command	REJECTED response
opcode	JOB QUEUE	40 ₁₆	←
operand[0]	subfunction	03 ₁₆ (close_job)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	job_position	FFFF ₁₆	←
operand[4]			
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	current_copies	FFFF ₁₆	←
operand[18]			
operand[19]	requested_copies	FFFF ₁₆	←
operand[20]			
operand[21]	current_pages	FFFF ₁₆	←
operand[22]			
operand[23]	requested_pages	FFFF ₁₆	←
operand[24]			
operand[25]	job_position	FFFF ₁₆	←
operand[26]			

For the failed close_job subfunction, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.14 – Possible result value for the rejected close_job subfunction

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified <i>print_job_ID</i> does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.

6.2.1.3 JOB QUEUE STATUS command

The JOB QUEUE STATUS command (ctype = STATUS) is used to get the current status of the specified job. The format of JOB QUEUE STATUS command format is illustrated by the figure below:

	length	msb						lsb
opcode	1	JOB QUEUE (40 ₁₆)						
operand[0]	1	FF ₁₆						
operand[1]	1	FF ₁₆						
operand[2]	1	FF ₁₆						
operand[3]	2	Job_position						
operand[4]								
operand[5]	12	print_job_ID						
:								
operand[16]								
operand[17]	2	FFFF ₁₆						
operand[18]								
operand[19]	2	FFFF ₁₆						
operand[20]								
operand[21]	2	FFFF ₁₆						
operand[22]								
operand[23]	2	FFFF ₁₆						
operand[24]								
operand[25]	2	FFFF ₁₆						
operand[26]								

Figure 6-4 – JOB QUEUE STATUS command format

The format of JOB QUEUE STATUS response format is illustrated by the figure below:

	length	msb						lsb
opcode	1	JOB QUEUE (40 ₁₆)						
operand[0]	1	FF ₁₆						
operand[1]	1	status						
operand[2]	1	result						
operand[3]	2	job_position						
operand[4]								
operand[5]	12	print_job_ID						
:								
operand[16]								
operand[17]	2	current_copies						
operand[18]								
operand[19]	2	requested_copies						
operand[20]								
operand[21]	2	current_pages						
operand[22]								
operand[23]	2	requested_pages						
operand[24]								
operand[25]	2	number_of_jobs						
operand[26]								

Figure 6-5 – JOB QUEUE STATUS response format

6.2.1.3.1 Field definitions

The field definitions of JOB QUEUE STATUS command and its response are same as the CONTROL command frame, as described in 6.2.1.2.1.

6.2.1.3.2 STATUS command operations

JOB QUEUE STATUS command is used to get information about the job. There are two ways to specify the job by JOB QUEUE STATUS command. One is to specify the *print_job_ID*, and the other is to specify the *job_position*.

The following clauses defines the detail of STATUS command and response. A controller shall issue the JOB QUEUE STATUS commands as described below.

6.2.1.3.2.1 Get the position in the queue and status of the specified job

The following Table 6.15 shows the sample command-response for a successful STATUS command with specified *print_job_ID* value. In this case, the *job_position* value shall be set to FFFF₁₆.

Table 6.15 – Successful command-response for STATUS command with specified print_job_ID

	opcode, operands	STATUS command	STABLE response
opcode	JOB QUEUE	40 ₁₆	←
operand[0]	FF ₁₆	FF ₁₆	←
operand[1]	status	FF ₁₆	<i>current job status</i>
operand[2]	result	FF ₁₆	00 ₁₆ (no_error)
operand[3]	job_position	FFFF ₁₆	<i>current job position</i>
operand[4]			
operand[5]	print_job_ID	controller EUI-64	←
:			
Operand[12]			
Operand[13]			
:	record ID		←
Operand[16]			
Operand[17]	current_copies	FFFF ₁₆	<i>current copies</i>
Operand[18]			
Operand[19]	requested_copies	FFFF ₁₆	<i>requested copies</i>
Operand[20]			
Operand[21]	current_pages	FFFF ₁₆	<i>current pages</i>
Operand[22]			
Operand[23]	requested_pages	FFFF ₁₆	<i>requested pages</i>
Operand[24]			
Operand[25]	number_of_jobs	FFFF ₁₆	<i>current number of jobs</i>
Operand[26]			

NOTE The *print_job_ID* value shall not be FFFFFFFFFFFFFFFFFFFFFFFFFF₁₆ when the *job_position* value had been FFFF₁₆. If this value had been set for the command frame, the target returns NOT IMPLEMENTED response.

The following Table 6.16 shows the sample command-response for a failed STATUS command with specified *print_job_ID* value.

Table 6.16 – Failed command-response for STATUS command with specified print_job_ID

	opcode, operands	STATUS command	REJECTED response
opcode	JOB QUEUE	40 ₁₆	←
operand[0]	FF ₁₆	FF ₁₆	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	job_ position	FFFF ₁₆	←
operand[4]			
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:			
operand[16]	current_copies	FFFF ₁₆	←
operand[17]			
operand[18]	requested_copies	FFFF ₁₆	←
operand[19]			
operand[20]	current_pages	FFFF ₁₆	←
operand[21]			
operand[22]	requested_pages	FFFF ₁₆	←
operand[23]			
operand[24]	number_of_jobs	FFFF ₁₆	←
operand[25]			
operand[26]			

For the failed STATUS command with the specified *print_job_ID* value, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.17 – Possible result value for the rejected STATUS command with specified print_job_ID

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified <i>print_job_ID</i> does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.

6.2.1.3.2.2 Get the print_job_ID and status of the job specified by its job_position

The following Table 6.18 shows the sample command-response for a successful STATUS command with specified *job_position* value.

Table 6.18 – Successful command-response for STATUS command with specified job_position

	opcode, operands	STATUS command	STABLE response
opcode	JOB QUEUE	40 ₁₆	←
operand[0]	FF ₁₆	FF ₁₆	←
operand[1]	status	FF ₁₆	<i>current job status</i>
operand[2]	result	FF ₁₆	00 ₁₆ (no_error)
operand[3]	job_position	<i>specified job position</i>	←
operand[4]			
operand[5]	Print_job_ID	FFFFFFFFFFFFFFFF ₁₆	<i>controller EUI-64</i>
:			
operand[12]			
operand[13]			
:			
operand[16]	current_copies	FFFF ₁₆	<i>current copies</i>
operand[17]			
operand[18]	requested_copies	FFFF ₁₆	<i>requested copies</i>
operand[19]			
operand[20]	current_pages	FFFF ₁₆	<i>current pages</i>
operand[21]			
operand[22]	requested_pages	FFFF ₁₆	<i>requested pages</i>
operand[23]			
operand[24]	number_of_jobs	FFFF ₁₆	<i>number of jobs</i>
operand[25]			
operand[26]			

NOTE The job_position value shall not be FFFF₁₆ when the *print_job_ID* value had been FFFFFFFFFFFFFFFFFFFFFFFFFF₁₆. If this value had been set for the command frame, the target returns NOT IMPLEMENTED response.

The following Table 6.19 shows the sample command-response for a failed STATUS command with specified *job_position* value.

Table 6.19 – Failed command-response for the get_job subfunction with job_position

	opcode, operands	STATUS command	REJECTED response
opcode	JOB QUEUE	40 ₁₆	←
operand[0]	FF ₁₆	FF ₁₆	←
operand[1]	job_status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	job_position	<i>specified job position</i>	←
operand[4]			
operand[5]	print_job_ID	FFFFFFFFFFFFFFFF ₁₆	←
:			
operand[12]			
operand[13]			
:			
operand[16]	current_copies	FFFF ₁₆	←
operand[17]			
operand[18]	requested_copies	FFFF ₁₆	←
operand[19]			
operand[20]	current_pages	FFFF ₁₆	←
operand[21]			
operand[22]	requested_pages	FFFF ₁₆	←
operand[23]			
operand[24]	number_of_jobs	FFFF ₁₆	←
operand[25]			
operand[26]			

For the failed STATUS command with specified *job_position* value, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.20 – Possible result value for the rejected STATUS command

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified <i>job_position</i> does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.

6.2.2 OPERATION MODE command

OPERATION MODE command is used to set or get information about the operation mode of the job.

6.2.2.1 OPERATION MODE CONTROL command

OPERATION MODE CONTROL command (ctype = CONTROL) is used to set information about the operation mode of the printer subunit for the specified job. The format of OPERATION MODE CONTROL command is illustrated by the figure below:

	length	msb						lsb
opcode	1	OPERATION MODE (41 ₁₆)						
operand[0]	2	reserved						
operand[1]								
operand[2]	1	result						
operand[3]	2	reserved						
operand[4]								
operand[5]	12	print_job_ID						
:								
operand[16]								
operand[17]	12	operation_mode_parameters						
:								
operand[28]								

Figure 6-6 – OPERATION MODE CONTROL command

6.2.2.1.1 Field definitions

result: This field shall be set to FF₁₆ in the CONTROL command frame. In case of the response frame of a CONTROL command, this field indicates the execution result of the CONTROL command. The detailed definition of this field is described in Table 6.21 as follows.

Table 6.21 – result field definitions

Value	Symbol	Meaning
00 ₁₆	no_error	The command execution has been finished successfully with no errors.
86 ₁₆	no_job_exists	The print job with specified <i>print_job_ID</i> does not exist.
87 ₁₆	Busy	The command is rejected because the subunit is busy.
89 ₁₆	illegal_sequence	The command is rejected because the job state is not opened_idle.
8A ₁₆	unsupported_parameter	The command is rejected because the specified parameter is not supported by the subunit.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.
other values	-	Reserved

NOTE The OPERATON MODE CONTROL command can be accepted by the printer subunit when the specified job is opened_idle state. Otherwise the OPERATION MODE CONTROL command is rejected with the result value set to 89₁₆ (illegal_sequence).

print_job_ID: Identifier of the print job requested by the controller or the subunit itself, defined in Table 6.5.

operation_mode_parameters: Parameter data for the printer job operation, as defined in Table 6.22 as follows. The details of these parameters are described in “6.2.2.3 Operation_mode_parameters detail”.

Table 6.22 – operation_mode_parameters definition

Address Offset	Contents
00 ₁₆	multiple_tiled
01 ₁₆	reserved
02 ₁₆	number_of_pics
03 ₁₆	
04 ₁₆	number_of_copies
05 ₁₆	
06 ₁₆	
07 ₁₆	speed_quality
08 ₁₆	mono_color
09 ₁₆	sizing
0A ₁₆	orientations
0B ₁₆	posx
	posy

6.2.2.1.2 OPERATION MODE CONTROL command operations

The following clauses define the detail of CONTROL command and response. A controller shall issue the OPERATION MODE CONTROL command as described below.

The following Table 6.23 shows the sample command-response for a successful OPERATION MODE CONTROL command.

Table 6.23 – Successful command-response for the OPERATION MODE CONTROL command

	opcode, operands	CONTROL command	ACCEPTED response
opcode	OPERATION MODE	41 ₁₆	←
operand[0]	reserved	0000 ₁₆	←
operand[1]			
operand[2]	result	FF ₁₆	00 ₁₆ (no_error)
operand[3]	reserved	000000 ₁₆	←
operand[4]			
operand[5]	print_job_ID	controller EUI-64	←
:			
operand[12]			
operand[13]			
:	record ID	←	
operand[16]			
operand[17]	operation_mode_parameters	requested parameters	←
:			
operand[28]			

The following Table 6.24 shows the sample command-response for a failed OPERATION MODE CONTROL command.

Table 6.24 – Failed command-response for the OPERATION MODE CONTROL command

	opcode, operands	CONTROL command	REJECTED response
opcode	OPERATION MODE	41 ₁₆	←
operand[0]	reserved	0000 ₁₆	←
operand[1]			
operand[2]	result	FF ₁₆	result value
operand[3]	reserved	0000 ₁₆	←
operand[4]			
operand[5]	print_job_ID	controller EUI-64	←
:			
operand[12]			
operand[13]			
:	record ID	←	
operand[16]			
operand[17]	operation_mode_parameters	requested parameters	←
:			
operand[28]			

For the failed OPERATION MODE CONTROL command, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.25 – Possible result value for the rejected OPERATION MODE CONTROL command

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified <i>print_job_ID</i> does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
89 ₁₆	illegal_sequence	The command is rejected because the job state is not opened_idle.
8A ₁₆	unsupported_parameter	The command is rejected because the specified parameter is not supported by the subunit.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.

NOTE If the values of the *operation_mode_parameters* are not supported by the printer subunit, it returns the REJECTED response with the *result* value set to unsupported_parameter. If other fields than *operation_mode_parameters* are not supported by the printer subunit, it returns the NOT IMPLEMENTED response.

6.2.2.2 OPERATION MODE STATUS command

OPERATION MODE STATUS command (ctype = STATUS) is used to get information about the operation mode of the printer subunit for the specified job. The format of OPERATION MODE STATUS command is illustrated by the figure below:

	length	msb							lsb
opcode	1	OPERATION MODE (41 ₁₆)							
operand[0]	1	subfunction							
operand[1]	1	FF ₁₆							
operand[2]	1	FF ₁₆							
operand[3]	2	FFFF ₁₆							
operand[4]									
operand[5]	12	print_job_ID							
:									
operand[16]									
operand[17]	12	operation_mode_parameters							
:									
operand[28]									

Figure 6-7 – OPERATION MODE STATUS command format

The following Figure 6-8 indicates the response frame format for the OPERATION MODE STATUS command.

	length	msb						lsb
opcode	1	OPERATION MODE (41 ₁₆)						
operand[0]	1	subfunction						
operand[1]	1	00 ₁₆						
operand[2]	1	result						
operand[3]	2	0000 ₁₆						
operand[4]								
operand[5]	12	print_job_ID						
:								
operand[16]								
operand[17]	12	operation_mode_parameters						
:								
operand[28]								

Figure 6-8 – OPERATION MODE STATUS response format

6.2.2.2.1 Field definitions

subfunction: This field indicates what kind of information a controller requests. The values of this field are defined as follows.

Table 6.26 – subfunction field definitions

Value	Symbol	Meaning
00 ₁₆	query_support_level	Query the specified operation mode parameters are supported by the printer subunit.
FF ₁₆	get_current_parameter	Get the current operation mode parameters for the specified <i>print_job_ID</i> .
other values	-	Reserved

If the value of the *subfunction* field is *query_support_level*, the *print_job_ID* field shall be FFFFFFFFFFFFFFFFFFFFFFFFFF₁₆, and the *operation_mode_parameters* shall not be filled with FF₁₆ for the STATUS command frame and response frame. The response frame for a STATUS command with the subfunction of *query_support_level*, the result field indicates the specified *operation_mode_parameters* is supported by the subunit or not.

If the value of the subfunction field is *get_current_parameter*, the *print_job_ID* field shall specify the ID value of the job in the queue, and the *operation_mode_parameters* shall be filled with FF₁₆, for the STATUS command frame. In the response frame for a STATUS command with the *subfunction* of *get_current_parameter*, the *operation_mode_parameters* field indicates the current operation mode parameter values.

Other field definitions of OPERATION MODE STATUS command and its response are same as the CONTROL command frame, as described in 6.2.2.1.1

6.2.2.2.2 OPERATION MODE STATUS command operations

OPERATION MODE STATUS command is used to get information about the operation mode of the printer subunit. There are two usage of OPERATION MODE STATUS command. One is to get the current operation mode values of the job specified by *print_job_ID*, and the other is to get the support level of the specified operation mode values.

The following clauses defines the detail of STATUS command and response. A controller shall issue the JOB QUEUE STATUS commands as described below.

6.2.2.2.1 Get_current_parameter subfunction

The following Table 6.27 shows the sample command-response for a successful OPERATION MODE STATUS command with the *subfunction* of get_current_parameter.

Table 6.27 – Successful command-response for the OPERATION MODE STATUS command with the subfunction of get_current_parameter

	opcode, operands	STATUS command	STABLE response
Opcode	OPERATION MODE	41 ₁₆	←
operand[0]	subfunction	FF ₁₆ (get_current_parameter)	←
operand[1]	-	FF ₁₆	00 ₁₆
operand[2]	result	FF ₁₆	00 ₁₆ (no_error)
operand[3]	-	FFFF ₁₆	0000 ₁₆
operand[4]	-	FFFF ₁₆	0000 ₁₆
operand[5]	print_job_ID	<i>controller</i> <i>EUI-64</i>	←
:			
operand[12]		<i>record ID</i>	←
operand[13]			
:			
operand[16]	operation_mode_parameters	FFFF... ₁₆	<i>current</i> <i>operation_mode_parameters</i>
operand[17]			
:			
operand[28]			

The following Table 6.28 shows the sample command-response for a failed OPERATION MODE STATUS command with the *subfunction* of get_current_parameter.

Table 6.28 – Failed command-response for the OPERATION MODE STATUS command with the subfunction of get_current_parameter

	opcode, operands	STATUS command	REJECTED response
opcode	OPERATION MODE	41 ₁₆	←
operand[0]	subfunction	FF ₁₆ (get_current_parameter)	←
operand[1]	-	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	-	FFFF ₁₆	←
operand[4]	-	FFFF ₁₆	←
operand[5]	print_job_ID	<i>controller</i> <i>EUI-64</i>	←
:			
operand[12]		<i>record ID</i>	←
operand[13]			
:			
operand[16]	operation_mode_parameters	FFFF... ₁₆	←
operand[17]			
:			
operand[28]			

For the failed OPERATION MODE STATUS command with the *subfunction* of get_current_parameter, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.29 – Possible result value for the rejected OPERATION MODE STATUS command with the subfunction of get_current_parameter

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified <i>print_job_ID</i> does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.

6.2.2.2.2 Query_support_level subfunction

The following Table 6.30 shows the sample command-response for a successful OPERATION MODE STATUS command with the *subfunction* of query_support_level. If the printer subunit supports the requested *operation_mode_parameters* values specified in the OPERATON MODE STATUS command with the *subfunction* of query_support_level, it returns STABLE response with the *result* value of no_error, otherwise the result value of unsupported_parameter.

Table 6.30 – Successful command-response for the OPERATION MODE STATUS command with the subfunction of query_support_level

	opcode, operands	STATUS command	STABLE response
opcode	OPERATION MODE	41 ₁₆	←
operand[0]	subfunction	00 ₁₆ (query_support_level)	←
operand[1]	-	FF ₁₆	00 ₁₆
operand[2]	result	FF ₁₆	<i>result</i>
operand[3]	-	FFFF ₁₆	0000 ₁₆
operand[4]	-	FFFF ₁₆	0000 ₁₆
operand[5]	print_job_ID	FFFFFFFFFFFFFFFF ₁₆	←
:			
operand[12]			
operand[13]			
:		FFFFFFFF ₁₆	←
operand[16]			
operand[17]	operation_mode_parameters	<i>requested operation_mode_parameters</i>	←
:			
operand[28]			

For the successful OPERATION MODE STATUS command with the *subfunction* of query_support_level, the *result* field of the STABLE response frame contains the result value, as follows.

Table 6.31 – Possible result value for the STABLE response for OPERATION MODE STATUS command with the subfunction of query_support_level

Value	Symbol	Meaning
00 ₁₆	no_error	The values of the requested <i>operation_mode_parameters</i> are supported by the subunit.
8A ₁₆	unsupported_parameter	The values of the requested <i>operation_mode_parameters</i> are not supported by the subunit.

The following Table 6.32 shows the sample command-response for a failed OPERATION MODE STATUS command with the *subfunction* of query_support_level.

Table 6.32 – Failed command-response for the OPERATION MODE STATUS command

	opcode, operands	STATUS command	REJECTED response
opcode	OPERATION MODE	41 ₁₆	←
operand[0]	subfunction	00 ₁₆ (query_support_level)	←
operand[1]	-	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	-	FFFF ₁₆	←
operand[4]			
operand[5]			
:			
operand[12]	print_job_ID	FFFFFFFFFFFFFFFF ₁₆	←
operand[13]			
:			
operand[16]	operation_mode_parameters	requested operation_mode_parameters	←
operand[17]			
:			
operand[28]			

For the failed OPERATION MODE STATUS command, the *result* field of the REJECTED response frame contains the result to indicate why the command had been rejected, as follows.

Table 6.33 – Possible result value for the rejected OPERATION MODE STATUS command

Value	Symbol	Meaning
87 ₁₆	busy	The command is rejected because the subunit is busy.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.

6.2.2.3 Operation_mode_parameters detail

This section defines the member of *operation_mode_parameters*.

6.2.2.3.1 No preference value

For every field except for *number_of_pics* and *number_of_copies*, the value of FF₁₆ indicates “don’t care” for the STATUS command frame with the subfunction of query_support_level, or for the CONTROL command frame.

For the CONTROL command, the value of FF₁₆ indicates “no preference for this field” and the target may select the appropriate value setting. The controller may investigate the current value for this field by STATUS command with the *subfunction* of get_current_parameter.

For *number_of_copies*, the value of FFFF₁₆ is reserved and shall not be used for the CONTROL command frame and STATUS command frame with the *subfunction* of query_support_level.

6.2.2.3.2 Device_default value

For every field except for *number_of_pics*, *number_of_copies*, and *multiple_tiled*, the value of zero is defined as “device_defaults”. If the field value is set to device_defaults, the result of the operation is printer dependent and is not predictable. Furthermore, the result may be different from other behaviors listed in the tables.

For example, a "simple" printer may support "device defaults" only, and it may have only limited capabilities (e.g. portrait only for orientation).

On the other hand, a "smart" printer which has more sophisticated feature, may support not only all the behaviors listed in the tables but also device_default whose result will be more "smart" and "flexible" behavior than other behaviors listed in the table, and it may do more intelligent operations (e.g. rotate image according to its aspect for orientation).

6.2.2.3.3 Multiple_tiled

This field indicates how the multiple print contents are going to be laid out on a page, defined as follows.

Table 6.34 – Definitions of multiple_tiled

Value	Symbol	Meaning
00 ₁₆	not_tiled	The following <i>number_of_pics</i> field indicates how many print contents will be laid out on a page. This value is device_default. (Mandatory)
01 ₁₆	tiled	The following <i>number_of_pics</i> field indicates how many times the one print content will be laid out on a page. (Optional)
Other values	-	Reserved

If the tiled value is set, the following *number_of_pics* field indicates how many times the one print content will be laid out on a page.

For example, if this field is set to 01₁₆ (tiled), and the *number_of_pics* is set to 9, after a single print content had been captured by the printer subunit, the same print content will be printed out 9 times on a single page.

If this field is set to 00₁₆ (not_tiled), the following *number_of_pics* field indicates how many (independent) print contents will be laid out on a page.

6.2.2.3.4 Number_of_pics

This field indicates how many print contents will be laid out per output page.

NOTE If the previous field (multiple_tiled field) had been set to 01₁₆ (tiled), the value of this field does not represent the number of valid print content sets to be actually transferred, but the value of this field is the number of print contents the printer subunit assumes to lay out in one page. If the previous field (multiple_tiled field) had been set to 00₁₆, print content captured by the printer subunit for one page is assumed to complete when the print contents are captured an equal number of times as the specified value for this field. The JOB QUEUE CONTROL command with the subfunction of close_job, also completes the one page. to be put in a queue. For example, in the case of the multiple_tiled field being set to 00₁₆ (not_tiled), when the printer subunit supports a *number_of_pics* value of 9, and the value of this field is set to 9 by OPERATION MODE CONTROL command, the printer subunit will either expect CAPTURE CONTROL command 9 times, or JOB QUEUE CONTROL command with the subfunction of close_job.

A minimum number of one is subject to mandatory support by the target device.

The value of FFFF₁₆ is reserved for the command frame, and the value of 0000₁₆ shall not be used.

The controller should not expect the printer subunit to support all the values from 1 to maximum number of *number_of_pics* value.

The controller may check to see that the printer subunit supports the specified *number_of_pics* value by OPERATION MODE STATUS command with the *subfunction* of *query_support_level*. The return value for this field of FFFF₁₆ indicate that this value is not supported.

6.2.2.3.5 Number_of_copies

This field indicates the number of copies for a single page to be printed.

A minimum number of one is subject to mandatory support by the target, and the value of FFFF₁₆ is reserved for the command frame, and the value of 0000₁₆ shall not be used.

6.2.2.3.6 Speed_quality

This field indicates the print output quality and speed, defined as follows.

Table 6.35 – Definitions of speed_quality

Value	Symbol	Meaning
00 ₁₆	device_default	The print output quality is set as device default. (Mandatory)
01 ₁₆	quality	Priority on print content quality. The printer might be processed slower, high print quality. (Optional)
02 ₁₆	speed	Priority on speed. The printer will be processed faster, lower print quality. (Optional)
Other values	-	Reserved

6.2.2.3.7 Mono_color

This field indicates the print color, specifying either of black-and-white, grayscale or color as the print output. If the controller specifies black-and-white or grayscale, the result shall be always in black-and-white or grayscale even if the original content is color.

Table 6.36 – Definitions of mono_color

Value	Symbol	Meaning
00 ₁₆	Device_default	The print output color is set as device default. (Mandatory)
01 ₁₆	color	The print output is color. (Optional)
02 ₁₆	grayscale	The print output is grayscale. (Optional)
03 ₁₆	black_and_white	The print output is black-and-white. (Optional)
Other values	-	Reserved

6.2.2.3.8 Sizing

This field indicates the output sizing of the print, defined as follows.

Table 6.37 – Definitions of sizing

Value	Symbol	Meaning
00 ₁₆	device_default	The print output will be sized as device default. (Mandatory)
01 ₁₆	small	The print output will be sized small, relative to the "medium" sizing within the output media without changes to aspect ratio of the print content. (Optional)
02 ₁₆	medium	The print output will be sized smaller than the "large_no_clip" sizing within the output media without changes to aspect ratio of the print content. (Optional)
03 ₁₆	large_no_clip	The print output will be sized, preserving aspect ratio without clipping, to fill as much of the output media as possible. (Optional)
04 ₁₆	large_with_clip	The print output will be sized, preserving aspect ratio and clipping if necessary, to completely fill the output media. (Optional)
Other values	-	Reserved

6.2.2.3.9 Orientations

This field indicates the output orientation of print output, defined as follows.

Table 6.38 – Definitions of orientations

Value	Symbol	Meaning
00 ₁₆	device_default	The raster direction of the image data is device default. (Mandatory)
01 ₁₆	portrait	The raster direction of the image data corresponds to the short dimension of the output media. (Optional)
02 ₁₆	landscape	The raster direction of the image data corresponds to the long dimension of the output media. (Optional)
03 ₁₆	mirrored_portrait	The horizontally mirrored image of Portrait. (Optional)
04 ₁₆	mirrored_landscape	The horizontally mirrored image of Landscape. (Optional)
Other values	-	Reserved

6.2.2.3.10 Posx

This field indicates the output X direction (the shorter dimension of the media) positioning of the print output, defined as follows.

Table 6.39 – Definitions of posx

Value	Symbol	Meaning
00 ₁₆	device_default	The image output will be positioned device default. (Mandatory)
01 ₁₆	center	The image output will be positioned at the center in the X direction. (Optional)
02 ₁₆	left	The image output will be positioned towards the left side in the X direction. (Optional)
03 ₁₆	right	The image output will be positioned towards the right side in the X direction. (Optional)
Other values	-	Reserved

6.2.2.3.11 Posy

This field indicates the output Y direction (the longer dimension of the media) positioning of the print output, defined as follows.

Table 6.40 – Definitions of posy

Value	Symbol	Meaning
00 ₁₆	device_default	The image output will be positioned device default. (Mandatory)
01 ₁₆	center	The image output will be positioned at the center in the Y direction. (Optional)
02 ₁₆	top	The image output will be positioned towards the top in the Y direction. (Optional)
03 ₁₆	bottom	The image output will be positioned towards the bottom in the Y <i>direction</i> . (Optional)
Other values	-	Reserved

6.2.2.4 Default values of operation_mode_parameters

The following Table 6.41 shows the default value of *operation_mode_parameters*.

Table 6.41 – Default value of operation_mode_parameters

Address Offset	Contents	Default value
00 ₁₆	multiple_tiled	00 ₁₆
01 ₁₆	reserved	00 ₁₆
02 ₁₆ - 03 ₁₆	number_of_pics	0001 ₁₆
04 ₁₆ - 05 ₁₆	number_of_copies	0001 ₁₆
06 ₁₆	speed_quality	00 ₁₆
07 ₁₆	mono_color	00 ₁₆
08 ₁₆	sizing	00 ₁₆
09 ₁₆	orientations	00 ₁₆
0A ₁₆	posx	00 ₁₆
0B ₁₆	posy	00 ₁₆

6.2.3 CAPTURE command

CAPTURE command is used to receive an image to the specified job, and get information about capabilities of the printer subunit for image receipt.

6.2.3.1 CAPTURE CONTROL command

CAPTURE CONTROL command (ctype = CONTROL) is used to receive an image from the asynchronous plug, or from the isochronous plug, or from an /external plug, or a source plug of other subunits inside the AV unit. The format of CAPTURE CONTROL command is illustrated by the figure below.

	length	msb						lsb
opcode	1	CAPTURE (42 ₁₆)						
operand[0]	1	subfunction						
operand[1]	1	status						
operand[2]	1	result						
operand[3]	1	destination_plug						
operand[4]	1	reserved						
operand[5]	12	print_job_ID						
:								
operand[16]								
operand[17]	2	image_format_specifier						
operand[18]								
operand[19]	2	reserved						
operand[20]								
operand[21]	4	data_size						
:								
operand[24]								
operand[25]	2	image_size_x						
operand[26]								
operand[27]	2	image_size_y						
operand[28]								
operand[29]	2	next_pic						
operand[30]								
operand[31]	2	next_page						
operand[32]								

Figure 6-9 – CAPTURE CONTROL command basic format

6.2.3.1.1 Field definitions

subfunction: This field indicates the operation to be taken by the target as defined in the Table 6.42 below.

Table 6.42 – subfunction field definitions for CAPTURE CONTROL command

Value	Symbol	Meaning
01 ₁₆	receive	Capture an image
02 ₁₆	abort	Abort capturing an image
04 ₁₆	resume	resume capturing an image after a serial bus reset
80 ₁₆	query	Get the supported values (e.g. <i>data_size</i> , <i>image_size_x</i> , <i>image_size_y</i>) for the specified <i>image_format_specifier</i> (for STATUS command)
FF ₁₆	get_status	Get the operation status (for STATUS command)
Other values	-	reserved

status: In case of the response frame of the CONTROL command or the STATUS command, this field indicates the state of the target, as defined in the Table 6.43. In case of the CONTROL command frame or STATUS command frame, this field shall be set to FF₁₆.

Table 6.43 – status field definitions

Value	Symbol	Meaning
00 ₁₆	idle	The command execution had been finished successfully, or no command is under execution.
01 ₁₆	Receiving_processing	The target is now receiving the data and processing it.
02 ₁₆	resume_wait	The serial bus reset occurred, the target is waiting to be resumed.
FF ₁₆	-	No meaning
Other values	-	Reserved

result: This field indicates the result of the command execution in case of the response frame of the CONTROL command, as defined in the Table 6.44. In case of the CONTROL command frame or STATUS command frame, this field shall be set to FF₁₆.

Table 6.44 – result field definitions

Value	Symbol	Meaning
00 ₁₆	no_error	The command execution had been finished successfully, or no command is under execution.
80 ₁₆	format_error	The command execution had been aborted, because the image format is not supported or unacceptable now.
81 ₁₆	image_size_error	The command execution had been aborted, because the image size is not supported or unacceptable now.
82 ₁₆	data_size_error	The command execution had been aborted, because the data length is not supported or unacceptable now.
83 ₁₆	invalid_format	The command execution had been aborted, because the image format is not the same as specified by the command frame
84 ₁₆	invalid_image_size	The command execution had been aborted, because the image size is not the same as specified by the command frame
85 ₁₆	invalid_data_size	The command execution had been aborted, because the receiving data had been lost or the receiving data is bigger than specified.
86 ₁₆	no_job_exists	The print job with specified <i>print_job_ID</i> does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy
88 ₁₆	no_process_to_abort	The command is rejected because currently there is no capturing process undergoing.
90 ₁₆	no_pending_process	The resume subfunction is rejected, because there is no pending capturing process. The capturing may be finished.
91 ₁₆	aborted	The command is aborted because the controller issued the CAPTURE CONTROL command with the <i>subfunction</i> of abort.
92 ₁₆	queue_full	The command is rejected because the job queue is full.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred
FF ₁₆	-	No meaning
Other values	-	Reserved

destination_plug: This field indicates the destination plug number (00₁₆ - 1E₁₆) of this subunit to input the image . If the specified plug number is not supported by the printer subunit, it returns NOT IMPLEMENTED response.

print_job_ID: This field specifies the print job entry, as defined in the Table 6.5.

data_size: For the CONTROL command, this field specifies the number of data bytes that would be transferred to the subunit from the asynchronous input plug of the AV unit. If the subunit plug is connected to the other unit plug than asynchronous input plug, or to the other subunit source plug, this field has no meaning and the controller shall set FFFFFFFF₁₆ for this field.

If the controller want to generate the blank area (without capturing an image data), this field shall be set to 00000000₁₆ and the *image_size_x* and *image_size_y* shall be set to 0000₁₆, and the *imaget_format_specifier* value shall be set to FE₁₆ (don't care).

Table 6.45 – data size field definitions

The data source	Value of the data_size field for the CONTROL command frame
source plug of the other subunit within the AV unit	FFFFFFFF ₁₆
asynchronous plug of the AV unit	The length of the data in bytes to be transferred to the subunit
isochronous plug of the AV unit	FFFFFFFF ₁₆
external plug of the AV unit	FFFFFFFF ₁₆

For the response frame of the STATUS command with the *subfunction* of *get_status*, this field indicates how much data bytes had been received by the subunit if the image source had been the asynchronous plug of the AV unit. In this case, the value of FFFFFFFF₁₆ is reserved and has no meaning.

For the response frame of the STATUS command with the *subfunction* of *query*, this field indicates the maximum data size for the specific image format specified by *imaget_format_specifier*.

If the image source had NOT been the asynchronous plug of the AV unit, this field has no meaning and the value of FFFFFFFF₁₆ is returned.

image_size_x: this field indicates the number of pixels for X-axis of the receiving image data. When the controller want to generate the blank area (without capturing an image data), this field shall be set to 0000₁₆. If the *imaget_format_specifier* had NOT been 8000₁₆ ~ 8FFF₁₆ or FE00₁₆~ FE01₁₆, this field shall not be FFFF₁₆ in the CONTROL command frame and the controller shall specify this field value.

image_size_y: This field indicates the number of pixels for Y-axis of the receiving image data. When the controller want to make the blank area (without capturing an image data), this field shall be set to 0000₁₆. If the *imaget_format_specifier* had NOT been 8000₁₆ ~ 8FFF₁₆ or FE00₁₆~ FE01₁₆, this field shall not be FFFF₁₆ in the CONTROL command frame and the controller shall specify this field value.

image_format_specifier: This field specifies the imageformat of the receiving image. The upper byte of this field indicates the category of the specified image format. The lower byte of this field specifies the detailed image format.

Table 6.46 – image_format_specifier definitions

Value(MSB)	Value(LSB)	Meaning
00₁₆	-	sRGB raw category
00 ₁₆	00 ₁₆	sRGB raw
00 ₁₆	01 ₁₆	sRGB raw, quadlet aligned
01₁₆	-	YCC raw category
01 ₁₆	0X ₁₆	YCC 4:2:2 raw, chunky
01 ₁₆	1X ₁₆	YCC 4:2:2 raw, linear
01 ₁₆	8X ₁₆	YCC 4:2:0 raw, chunky
01 ₁₆	9X ₁₆	YCC 4:2:0 raw, linear
01 ₁₆	X0 ₁₆	Pixel ratio 1.00 x 1.00 / ITU-R BT.709-4 / interlace
01 ₁₆	X1 ₁₆	Pixel ratio 1.19 x 1.00 / ITU-R BT.709-4 / interlace
01 ₁₆	X2 ₁₆	Pixel ratio 0.89 x 1.00 / ITU-R BT.709-4 / interlace
01 ₁₆	X3 ₁₆	Pixel ratio 0.89 x 1.00 / ITU-R BT.601-5 / interlace
01 ₁₆	X8 ₁₆	Pixel ratio 1.00 x 1.00 / ITU-R BT.709-4 / progressive
01 ₁₆	X9 ₁₆	Pixel ratio 1.19 x 1.00 / ITU-R BT.709-4 / progressive
01 ₁₆	XA ₁₆	Pixel ratio 0.89 x 1.00 / ITU-R BT.709-4 / progressive
01 ₁₆	XB ₁₆	Pixel ratio 0.89 x 1.00 / ITU-R BT.601-5 / progressive
10₁₆	-	JPEG File formats category
10 ₁₆	00 ₁₆	Exif 2.1, compressed file
10 ₁₆	01 ₁₆	JFIF
10 ₁₆	0F ₁₆	JPEG
10 ₁₆	10 ₁₆	Exif 2.1, uncompressed file
80₁₆ – 8F₁₆	-	Vender Dependent Formats category
80 ₁₆ – 8F ₁₆	00 ₁₆ – FF ₁₆	Vender Dependent Formats
FE ₁₆	00 ₁₆	Unit plug defined
FE ₁₆	01 ₁₆	Don't care
Other values		Reserved

NOTE The value of FE00₁₆ (Unit plug defined) is used when the data will be received from the isochronous serial bus plug. The value of FE01₁₆ (Don't care) is used when the data will be received from other subunit inside the same unit.

next_pic: This field indicates where the “next” image will be laid out on a page. If the *multiple_tiled* field is set to 00₁₆ (not_tiled), the minimum value of this field is zero, and the maximum value of this field is (*number_of_pic* – 1). If the *multiple_tiled* field is set to 01₁₆ (tiled), the value of this field shall be zero.

NOTE *number_of_picst* and *multiple_tiled* mentioned above is defined by *number_of_picst* value and *multiple_tiled* value specified by the *operation_mode_parameters* of the OPERATION MODE CONTROL command frame. (abbreviated as *number_of_picst* and *multiple_tiled* below)

For example, if the *multiple_tiled* field is set to 00_{16} (not_tiled) and the *number_of_picsnt* is set to 4, this field may vary from zero to 3.

For the command frame, this field shall be set to $FFFF_{16}$ and the response frame includes the current value of the target. Note that this field is updated when the execution of the CAPTURE CONTROL command with the *subfunction* of receive had been finished.

If the *multiple_tiled* field is set to 00_{16} (not_tiled) and the *number_of_picst* had been set to 4, and there is no CAPTURE CONTROL command with the *subfunction* of receive previously issued, this field value is zero in the STATUS response frame. When the CAPTURE CONTROL command with the *subfunction* of receive had been sent and the target starts capturing, this field value is still zero (which can be retrieved by STATUS command with the *subfunction* of get_status). When the final response for the CONTROL command (ACCEPTED or REJECTED) is returned from the target, this field may be one (a single picture had been captured successfully or unsuccessfully), or zero (a capturing had been failed completely).

next_page: This field indicates where the “next” page number. The minimum value of this field is zero. For the command frame, this field shall be set to $FFFF_{16}$ and the response frame includes the current value of the target in case of the CONTROL command or the STATUS command with the *subfunction* of get_status.

Note that this field is updated when the execution of the CAPTURE CONTROL command with the subfunction of receive had been finished when the *number_of_picst* had been set to one.

If there is no CAPTURE CONTROL command with the *subfunction* of receive previously issued, the STATUS response frame value is zero. When the CAPTURE CONTROL command with the *subfunction* of receive had been sent and the target starts capturing, this field value is still zero (which can be retrieved by STATUS command with the *subfunction* of get_status). When the final response for the CONTROL command (ACCEPTED or REJECTED) is returned from the target, this field may be one (a single picture had been captured successfully or unsuccessfully), or zero (a capturing had been failed completely).

Note that if the *number_of_pics* had been set to one, the *next_picst* value is always zero, and the *next_page* value is to be incremented.

6.2.3.1.2 CAPTURE CONTROL command operations

This section describes how the printer subunit behaves by the CAPTURE CONTROL commands.

6.2.3.1.2.1 CAPTURE CONTROL command state machine

The following shows the state transitions of the normal CAPTURE command operation. Note that this state machine only describes the state transitions triggered by the CAPTURE CONTROL commands or the serial bus reset or other events, and there is no description about other events like STATUS command or invalid command rejection.

CAPTURE operation state machine states are described below.

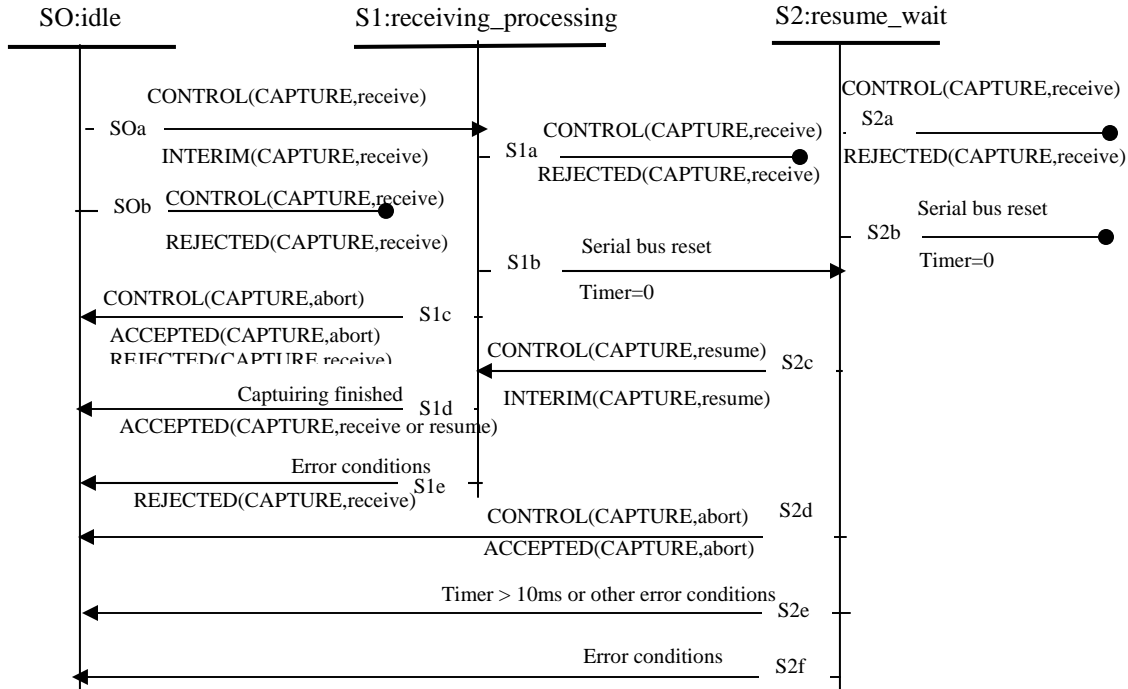


Figure 6-10 – CAPTURE CONTROL command operation state machine

S0: idle, The CAPTURE CONTROL command execution had been finished, or no CAPTURE CONTROL command is under execution.

S1: receiving_processing, The target is now receiving the data and / or processing it.

S2: resume_wait, The serial bus reset occurred, the target is waiting to be resumed.

6.2.3.1.2.1.1 State machine transitions

S0a. When the target receives the valid CAPTURE CONTROL command with the *subfunction* of receive, it returns the INTERIM response to the controller and transitions to the state S1.

S0b. When the target receives the invalid CAPTURE CONTROL command with the *subfunction* of receive or could not accept the command, it returns the REJECTED response to the controller.

S1a. When the target receives the additional CAPTURE CONTROL command with the *subfunction* of receive, it returns the REJECTED response to the controller, indicating the subunit is busy.

S1b. When the serial bus reset occurred, the target starts the timer to detect the resumption timeout, then transitions to the state S2.

S1c. When the target receives the valid CAPTURE CONTROL command with the *subfunction* of abort, it returns the ACCEPTED response, then returns REJECTED response for the previously issued CAPTURE CONTROL command with the *subfunction* of receive or resume, then transitions to the state S0.

S1d. When the target finished capturing process, it returns ACCEPTED response to the controller.

S1e. When the target detects some error conditions and stops capturing process, it returns the REJECTED response to the controller, then transitions to the state S0.

S2a. When the target receives the additional CAPTURE CONTROL command with the *subfunction* of receive, it returns the REJECTED response to the controller, indicating the subunit is busy.

S2b. When the serial bus reset occurred, the target restarts the timer to detect the resumption timeout.

S2c. When the target receives the valid CAPTURE CONTROL command with the *subfunction* of resume, it returns the ACCEPTED response, then transitions to the state S1.

S2d. When the target receives the valid CAPTURE CONTROL command with the *subfunction* of abort, it returns the ACCEPTED response, then transitions to the state S0.

S2d. When the target finished capturing process, it returns ACCEPTED response to the controller.

S2e. After the target detects the resumption timeout (10 seconds), it may stop capturing process, then transitions to the state S0.

S2f. When the target detects some error conditions and stops capturing process, transitions to the state S0.

6.2.3.1.2.2 Command resumption from serial bus reset

This section defines the behavior of the controller and target when a serial bus reset occurred. There are some scenarios for the bus-reset recovery.

(1) No influence by the serial bus reset (Internal connections, External plug connection)

When the subunit had been executing the capturing from the subunit inside the unit, or from the external plug, the serial bus reset only cancels the AV/C command response transactions, and the capturing process will be continued by the subunit. The controller should issue the resume subfunction to restart the AV/C command transaction.

(2) Asynchronous connections

When the bus reset occurred, the asynchronous connection transport stops its activities until the controller restores the connection. The controller should restore the asynchronous connection first, then issue the CAPTURE resume subfunction to restart the AV/C command transaction.

(3) Isochronous connections

When the bus reset occurred, the isochronous connection transport may not stop its activities, but the controller should restore the connection first, then issue the CAPTURE resume subfunction to restart the AV/C command transaction.

(4) Image source disappeared

After the serial bus reset, the imagesource unit might be disappeared from the bus. In this case, the controller could not restore the connection, so the controller should abort the CAPTURE command by abort subfunction.

6.2.3.1.2.2.1 Target behavior

The AV/C command transaction of CAPTURE CONTROL command is aborted by the serial bus reset. But the target is expected to continue the data receipt and processing if possible.

Until receipt of the CAPTURE CONTROL command with the subfunction of resume or 10 seconds timeout detected, the target should hold the previously issued CAPTURE CONTROL command information to prepare the resumption.

It is the controller's responsibility to restore the data transport path prior to the CAPTURE CONTROL command with the *subfunction* of resume. But if there is no data transport path restored when the target receives the CAPTURE CONTROL command with the *subfunction* of resume, the target rejects the resumption command.

6.2.3.1.2.2.2 Controller behavior

After the serial bus reset has been detected, the controller is expected to restore the data transport path (e.g. Asynchronous connection) first, then issue the CAPTURE CONTROL command which the *subfunction* of resume. As the target detects the resumption timeout after 10 seconds had passed from the latest serial bus reset, the controller should issue the resumption command within 10 seconds to the target. Even if the controller tried to restore the CAPTURE process, the CAPTURE operation may fail

The controller also may abort the capturing operation by CAPTURE CONTROL command with the *subfunction* of abort.

6.2.3.1.2.2.3 The final response loss

The final response of CAPTURE CONTROL command with the *subfunction* of receive may be lost by the serial bus reset. If the controller attempts to restore the capturing process which had already finished, the CAPTURE CONTROL command which the *subfunction* of resume, is rejected with the *result* field value of *no_pending_process*.

The controller can check the status of the printer subunit by CAPTURE STATUS command with the *subfunction* value set to FF₁₆. The *status* field include the current capturing status of the target, and the *next_picst* field and *next_page* field indicates where the next CAPTURE CONTROL command with the *subfunction* of receive stores the received data.

6.2.3.1.2.3 Executing the capturing operation

The CAPTURE CONTROL command with the *subfunction* of receive, directs the print job to receive the image data. The target returns ACCEPTED response for the CAPTURE CONTROL command with the *subfunction* of receive to the controller when the image data had been captured successfully. (If the target could not finish the capturing operation, it returns REJECTED response.)

The controller can expect the target is ready to process the next CAPTURE CONTROL command with the *subfunction* of receive.

The following Table 6.47 shows the sample command-response for a successful CAPTURE CONTROL command.

Table 6.47 – Successful command-response for the CAPTURE CONTROL command with the subfunction of receive

	opcode, operands	CONTROL command	ACCEPTED response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	01 ₁₆ (receive)	←
operand[1]	status	FF ₁₆	00 ₁₆ (idle)
operand[2]	result	FF ₁₆	00 ₁₆ (no error)
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	reserved	00 ₁₆	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:		<i>record ID</i>	←
operand[16]			
operand[17]	image_format_specifier	<i>image format</i>	←
operand[18]			
operand[19]	reserved	0000 ₁₆	←
operand[20]			
operand[21]	data_size	<i>input image data size (bytes)</i>	←
:			
operand[24]			
operand[25]	image_size_x	<i>image size (x) of input image (pixels)</i>	←
operand[26]			
operand[27]	image_size_y	<i>image size (y) of input image (pixels)</i>	←
operand[28]			
operand[29]	next_pict	FFFF ₁₆	<i>next picture number</i>
operand[30]			
operand[31]	next_page	FFFF ₁₆	<i>next page number</i>
operand[32]			

The following Table 6.48 shows the sample command-response transaction for a failed CAPTURE CONTROL command.

Table 6.48 – Failed command-response for CAPTURE CONTROL command with the subfunction of receive

	opcode, operands	CONTROL command	REJECTED response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	01 ₁₆ (receive)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	reserved	00 ₁₆	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	image_format_specifier	<i>image format</i>	←
operand[18]			
operand[19]	reserved	0000 ₁₆	←
operand[20]			
operand[21]			
:	data_size	<i>input iimage data size (bytes)</i>	←
operand[24]			
operand[25]			
operand[26]	image_size_x	<i>image size (x) of input image (pixels)</i>	←
operand[27]			
operand[28]			
operand[29]	image_size_y	<i>image size (y) of input image (pixels)</i>	←
operand[30]			
operand[31]			
operand[32]	next_pic	FFFF ₁₆	←
operand[31]	next_page	FFFF ₁₆	←
operand[32]			

NOTE When the target could not accept the command because the *image_format_specifier* or *data_size* or *image_size_x* or *image_size_y* values had not been supported, it returns REJECTED response with result. If other field of the CONTROL command frame had not been supported by the target, it returns NOT IMPLEMENTED response.

For the failed CAPTURE CONTROL command with the *subfunction* of receive, the *result* field of the REJECTED response field contains the result value to indicate why the command had been rejected, as follows.

Table 6.49 – Possible result value for the rejected receive subfunction

Value	Symbol	Meaning
80 ₁₆	format_error	The command execution had been aborted, because the image format is not supported or unacceptable now.
81 ₁₆	image_size_error	The command execution had been aborted, because the image size is not supported or unacceptable now.
82 ₁₆	data_size_error	The command execution had been aborted, because the data length is not supported or unacceptable now.
83 ₁₆	invalid_format	The command execution had been aborted, because the image format is not the same as specified by the command frame
84 ₁₆	invalid_image_size	The command execution had been aborted, because the image size is not the same as specified by the command frame
85 ₁₆	invalid_data_size	The command execution had been aborted, because the receiving data had been lost or the receiving data is bigger than specified.
86 ₁₆	no_job_exists	The print job with specified print_job_ID does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy
91 ₁₆	aborted	The command is aborted because the controller issued the CAPTURE CONTROL command with the subfunction of abort.
92 ₁₆	queue_full	The command is rejected because the job queue is full.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred
Other values	-	Reserved

6.2.3.1.2.4 Aborting the capturing operation after the serial bus reset

The controller can abort the undergoing capturing process by CAPTURE CONTROL command with the *subfunction* of abort.

The CONTROL command frame shall be the same as previously issued CAPTURE CONTROL command with the *subfunction* of receive, except for the subfunction field value.

If the CAPTURE CONTROL command with the *subfunction* of abort had been accepted by the target, it returns ACCEPTED response, and the target starts to stop the image data capturing operation. When the target had “almost” finished the previously issued CAPTURE CONTROL command with the subfunction of receive, the target may return REJECTED response for the abort *subfunction*, with the *result* of busy.

After the target returned the ACCEPTED response for the CAPTURE CONTROL command with the *subfunction* of abort, the target returns REJECTED response

After the target directs the print job to receive the image data. The target returns ACCEPTED response for the CAPTURE CONTROL command with the *subfunction* of resume to the controller when the image data had been captured successfully.

The following Table 6.50 shows the sample command-response for a successful CAPTURE CONTROL command with the subfunction of abort.

Table 6.50 – Successful command-response for the CAPTURE CONTROL command with the subfunction of abort

	opcode, operands	CONTROL command	ACCEPTED response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	02 ₁₆ (abort)	←
operand[1]	status	FF ₁₆	00 ₁₆ (idle)
operand[2]	result	FF ₁₆	00 ₁₆ (no error)
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	reserved	00 ₁₆	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	image_format_specifier	<i>image format</i>	←
operand[18]			
operand[19]	reserved	0000 ₁₆	←
operand[20]			
operand[21]	data_size	<i>input image data size (bytes)</i>	←
:			
operand[24]			
operand[25]	image_size_x	<i>image size (x) of input image (pixels)</i>	←
operand[26]			
operand[27]	image_size_y	<i>image size (y) of input image (pixels)</i>	←
operand[28]			
operand[29]	next_pic	FFFF ₁₆	<i>next picture number</i>
operand[30]			
operand[31]	next_page	FFFF ₁₆	<i>next page number</i>
operand[32]			

The following Table 6.51 shows the sample command-response transaction for a failed CAPTURE CONTROL command with the subfunction of abort.

Table 6.51 – Failed command-response for CAPTURE CONTROL command with the subfunction of abort

	opcode, operands	CONTROL command	REJECTED response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	02 ₁₆ (abort)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	reserved	00 ₁₆	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	image_format_specifier	<i>image format</i>	←
operand[18]			
operand[19]	reserved	0000 ₁₆	←
operand[20]	data_size	<i>input image data size (bytes)</i>	←
operand[21]			
:			
operand[24]	image_size_x	<i>image size (x) of input image (pixels)</i>	←
operand[25]			
operand[26]	image_size_y	<i>image size (y) of input image (pixels)</i>	←
operand[27]			
operand[28]	next_pic	FFFF ₁₆	←
operand[29]			
operand[30]	next_page	FFFF ₁₆	←
operand[31]			
operand[32]			

NOTE When the target could not accept the command because the *image_format_specifier* or *data_size* or *image_size_x* or *image_size_y* values had not been the same as the previously issued CONTROL command frame, it returns REJECTED response with the result of no_process_to_abort. If the specified *print_job_ID* is not supported by the target, it returns REJECTED response with the *result* of no_job_exists. If other field of the CONTROL command frame had not been supported by the target, it returns NOT IMPLEMENTED response.

For the failed CAPTURE CONTROL command with the subfunction of abort, the result field of the REJECTED response frame contains the *result* value to indicate why the command had been rejected, as follows.

Table 6.52 – Possible result value for the rejected abort subfunction

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified print_job_ID does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy
88 ₁₆	no_process_to_abort	The command is rejected because currently there is no capturing process undergoing.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred
Other values	-	Reserved

6.2.3.1.2.5 Restoring the capturing operation after the serial bus reset

The command response transaction of the CAPTURE CONTROL command with the *subfunction* of receive, is aborted by the serial bus reset, but the target may continue to process the data.

When the controller requests to restore the command response transaction, it issues the CAPTURE CONTROL command with the *subfunction* of resume. The CONTROL command frame shall be the same as previously issued CAPTURE CONTROL command with the *subfunction* of receive (or resume), except for the subfunction field value.

After the target directs the print job to receive the image data the target returns ACCEPTED response for the CAPTURE CONTROL command with the *subfunction* of resume to the controller when the image data had been captured successfully.

The following Table 6.53 shows the sample command-response for a successful CAPTURE CONTROL command with the subfunction of resume.

Table 6.53 – Successful command-response for the CAPTURE CONTROL command with the subfunction of resume

	opcode, operands	CONTROL command	ACCEPTED response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	04 ₁₆ (resume)	←
operand[1]	status	FF ₁₆	00 ₁₆ (idle)
operand[2]	result	FF ₁₆	00 ₁₆ (no error)
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	reserved	00 ₁₆	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]	image_format_specifier	<i>image format</i>	←
operand[17]			
operand[18]			
operand[19]	reserved	0000 ₁₆	←
operand[20]			
operand[21]	data_size	<i>input image data size (bytes)</i>	←
:			
operand[24]			
operand[25]	image_size_x	<i>image size (x) of input image (pixels)</i>	←
operand[26]			
operand[27]	image_size_y	<i>image size (y) of input image (pixels)</i>	←
operand[28]			
operand[29]	next_pic	FFFF ₁₆	<i>next picture number</i>
operand[30]			
operand[31]	next_page	FFFF ₁₆	<i>next page number</i>
operand[32]			

The following Table 6.54 shows the sample command-response transaction for a failed CAPTURE CONTROL command with the subfunction of resume.

Table 6.54 – Failed command-response for CAPTURE CONTROL command with the subfunction of resume

	opcode, operands	CONTROL command	REJECTED response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	04 ₁₆ (resume)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	reserved	00 ₁₆	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	image_format_specifier	<i>image format</i>	←
operand[18]			
operand[19]	reserved	0000 ₁₆	←
operand[20]	data_size	<i>input image data size (bytes)</i>	←
operand[21]			
:			
operand[24]			
operand[25]	image_size_x	<i>image size (x) of input image (pixels)</i>	←
operand[26]			
operand[27]	image_size_y	<i>image size (y) of input image (pixels)</i>	←
operand[28]			
operand[29]	next_pic	FFFF ₁₆	←
operand[30]	next_page	FFFF ₁₆	←
operand[31]			
operand[32]			

NOTE When the target could not accept the command because the *image_format_specifier* or *data_size* or *image_size_x* or *image_size_y* values had not been supported, it returns REJECTED response with the result of no_pending_process. If the specified *print_job_ID* is not supported by the target, it returns REJECTED response with the *result* of no_job_exists.

If other field of the CONTROL command frame had not been supported by the target, it returns NOT IMPLEMENTED response.

For the failed CAPTURE CONTROL command with the subfunction of resume, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.55 – Possible result value for the rejected resume subfunction

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified print_job_ID does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy
90 ₁₆	no_pending_process	The resume subfunction is rejected, because there is no pending capturing process. The capturing may be finished.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred
Other values	-	Reserved

6.2.3.2 CAPTURE STATUS command

The CAPTURE STATUS command (ctype=STATUS) is used to retrieve the information about the capturing operations.

6.2.3.2.1 Get the current capturing status

The get_status *subfunction* for the CAPTURE STATUS command is used to retrieve the current capture operation status.

The following Figure 6-11 indicates the command frame format of the CAPTURE STATUS command with the *subfunction* of get_status.

	length	msb						lsb
opcode	1	CAPTURE (42 ₁₆)						
operand[0]	1	subfunction						
operand[1]	1	FF ₁₆						
operand[2]	1	FF ₁₆						
operand[3]	1	FF ₁₆						
operand[4]	1	FF ₁₆						
operand[5]	12	print_job_ID						
:								
operand[16]	2	image_format_specifier						
operand[17]								
operand[18]	2	FFFF ₁₆						
operand[19]								
operand[20]	4	FFFFFFFF ₁₆						
operand[21]								
:								
operand[24]								
operand[25]	2	FFFF ₁₆						
operand[26]								
operand[27]	2	FFFF ₁₆						
operand[28]								
operand[29]	2	FFFF ₁₆						
operand[30]								
operand[31]	2	FFFF ₁₆						
operand[32]								

Figure 6-11 – CAPTURE STATUS command frame with subfunction get_status

The following Figure 6-12 indicates the response frame format of the CAPTURE STATUS command with the *subfunction* of get_status.

	length	msb						lsb
opcode	1	CAPTURE (42 ₁₆)						
operand[0]	1	subfunction						
operand[1]	1	status						
operand[2]	1	result						
operand[3]	1	destination_plug						
operand[4]	1	00 ₁₆						
operand[5]	12	print_job_ID						
:								
operand[16]								
operand[17]	2	image_format_specifier						
operand[18]								
operand[19]	2	0000 ₁₆						
operand[20]								
operand[21]	4	data_size						
:								
operand[24]								
operand[25]	2	image_size_x						
operand[26]								
operand[27]	2	image_size_y						
operand[28]								
operand[29]	2	next_pic						
operand[30]								
operand[31]	2	next_page						
operand[32]								

Figure 6-12 – CAPTURE status response frame with subfunction get_status

The detail of the CAPTURE status command operation is described in “6.2.3.2.3 CAPTURE STATUS command operations”.

6.2.3.2.1.1 Field definitions

subfunction: This field indicates the operation to be taken by the target as defined in the Table 6.56 below.

Table 6.56 – subfunction field definitions for CAPTURE STATUS command

Value	Symbol	Meaning
80 ₁₆	query	Get the supported values (e.g. data_size, image_size_x, image_size_y) for the specified image_format_specifier (for STATUS command)
FF ₁₆	get_status	Get the operation status (for STATUS command)
Other values	-	Reserved

print_job_ID: When a subfunction field value is query, this field value shall be FFFFFFFFFFFFFFFFFFFFFFFFFF₁₆ for both command and response frame. When a subfunction field value is get_status, this field value shall specify the print job in the queue for command frame.

image_format_specifier: When a subfunction field value is query, this field value shall specify the image format as defined in “Table 6.46 – image_format_specifier definitions” for both command and response frame. When the *subfunction* field value is get_status, this field value shall be FFFF₁₆ for command frame.

Other fields are defined according to the *subfunction* value, as illustrated below.

Table 6.57 – Definitions of other fields

Field	subfunction	Meaning
status	query	No meaning (FF ₁₆)
	get_status	Current status of capturing (see 6.2.3.1.2.1.1)
result	query	Command execution result
	get_status	
destination_plug	query	No meaning (FF ₁₆)
	get_status	Current destination_plug ID value
data_size	query	Maximum data_size for the specified image format
	get_status	Received number of bytes in case of capturing operation, otherwise no meaning (FFFFFFFF ₁₆)
image_size_x	query	Maximum image_size_x for the specified image format
	get_status	Specified value by the CAPTURE CONTROL command in case of capturing operation, otherwise no meaning (FFFF ₁₆)
image_size_y	query	Maximum image_size_y for the specified image format
	get_status	Specified value by the CAPTURE CONTROL command in case of capturing operation, otherwise no meaning (FFFF ₁₆)
next_pic	query	No meaning (FFFF ₁₆)
	get_status	Current value
next_page	query	No meaning (FFFF ₁₆)
	get_status	Current value

6.2.3.2.2 Get the maximum size for the specified print content_format

The query subfunction for the CAPTURE STATUS command is used to retrieve the maximum *data_size*, *image_size_x*, and *image_size_y* values for the specified *print_job_ID* and *image_format_specifier*.

The following Figure 6-13 indicates the command frame format of the CAPTURE STATUS command with the *subfunction* of query.

	length	msb						lsb
opcode	1	CAPTURE (42 ₁₆)						
operand[0]	1	80 ₁₆ (query)						
operand[1]	1	FF ₁₆						
operand[2]	1	FF ₁₆						
operand[3]	1	FF ₁₆						
operand[4]	1	FF ₁₆						
operand[5]	12	FFFFFFFFFFFFFFFF ₁₆						
:								
operand[16]								
operand[17]	2	image_format_specifier						
operand[18]								
operand[19]	2	FFFF ₁₆						
operand[20]								
operand[21]	4	FFFFFFFF ₁₆						
:								
operand[24]								
operand[25]	2	FFFF ₁₆						
operand[26]								
operand[27]	2	FFFF ₁₆						
operand[28]								
operand[29]	2	FFFF ₁₆						
operand[30]								
operand[31]	2	FFFF ₁₆						
operand[32]								

Figure 6-13 – CAPTURE STATUS command frame with subfunction query

The following Figure 6-14 indicates the response frame format of the CAPTURE STATUS command with the *subfunction* of query.

The CAPTURE STATUS command with the *subfunction* of query can be used whenever the specified print job is opened (the status of the specified print job is **S1:opened_idle** or **S2:opened_capturing**).

The detail of the CAPTURE status command operation is described in “6.2.3.2.3 CAPTURE STATUS command operations”.

	length	msb						lsb
opcode	1	CAPTURE (42 ₁₆)						
operand[0]	1	80 ₁₆ (query)						
operand[1]	1	FF ₁₆						
operand[2]	1	Result						
operand[3]	1	FF ₁₆						
operand[4]	1	00 ₁₆						
operand[5]	12	FFFFFFFFFFFFFFFF ₁₆						
:								
operand[16]								
operand[17]	2	image_format_specifier						
operand[18]								
operand[19]	2	0000 ₁₆						
operand[20]								
operand[21]	4	maximum data_size for the specified image format						
:								
operand[24]								
operand[25]	2	maximum image_size_x for the specified image format						
operand[26]								
operand[27]	2	maximum image_size_y for the specified image format						
operand[28]								
operand[29]	2	FFFF ₁₆						
operand[30]								
operand[31]	2	FFFF ₁₆						
operand[32]								

Figure 6-14 – CAPTURE status response frame with subfunction query

6.2.3.2.3 CAPTURE STATUS command operations

There are two *subfunction* values defined for the STATUS command operations.

6.2.3.2.3.1 Get the current capturing status

The *get_status subfunction* for the CAPTURE STATUS command is used to retrieve the current capture operation status.

The following Table 6.58 shows the sample command-response for a successful CAPTURE STATUS command with the *subfunction* of *get_status*.

Table 6.58 – Successful command-response for the CAPTURE STATUS command with the subfunction of get_status

	opcode, operands	STATUS command	STABLE response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	FF ₁₆ (get_status)	←
operand[1]	status	FF ₁₆	current status
operand[2]	result	FF ₁₆	00 ₁₆ (no error)
operand[3]	destination_plug	destination_plug number	←
operand[4]	-	FF ₁₆	00 ₁₆
operand[5]	print_job_ID	controller EUI-64	←
operand[12]			
operand[13]			
operand[16]		record ID	←
operand[17]	image_format_specifier	FFFF ₁₆	current image_format
operand[18]	-	FFFF ₁₆	0000 ₁₆
operand[19]	data_size	FFFFFFFF ₁₆	current data_size
operand[20]			
operand[21]	image_size_x	FFFF ₁₆	current image_size_x
operand[24]			
operand[25]			
operand[26]	image_size_y	FFFF ₁₆	current image_size_y
operand[27]			
operand[28]	next_pic	FFFF ₁₆	current next_pic
operand[29]			
operand[30]	next_page	FFFF ₁₆	current next_page
operand[31]			
operand[32]			

In the response frame, “current status”, “current image_format”, “current data_size”, “current image_size_x”, “current image_size_y”, “current next_pic”, and “current next_page” values are different according to the capturing status.

The following table indicates the parameter values that would be returned.

Table 6.59 – status response field values

Capturing state	current status	current image format	current data size	current image size x	current image size y
S0: idle	00 ₁₆	FFFF ₁₆	FFFFFFFF ₁₆	FFFF ₁₆	FFFF ₁₆
S1: receiving_processing	01 ₁₆	specified image format by the CONTROL command	Number of data bytes the printer subunit had been received in case that the CONTROL command frame specified the size other than FFFFFFFF ₁₆ ., otherwise FFFFFFFF ₁₆ .	specified image size specified by the CONTROL command	
S2: resume_wait	02 ₁₆				

The following Table 6.60 shows the sample command-response for a failed CAPTURE STATUS command with the subfunction of get_status.

Table 6.60 – Failed command-response for the CAPTURE STATUS command with the subfunction of get_status

	opcode, operands	STATUS command	REJECTED response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	FF ₁₆ (get_status)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	result value
operand[3]	destination_plug	destination_plug number	←
operand[4]	-	FF ₁₆	←
operand[5]	print_job_ID	controller EUI-64	←
:			
operand[12]		record ID	←
operand[13]			
:			
operand[16]	image_format_specifier	FFFF ₁₆	←
operand[17]			
operand[18]	-	FFFF ₁₆	←
operand[19]			
operand[20]	data_size	FFFFFFFF ₁₆	←
operand[21]			
:			
operand[24]			
operand[25]	image_size_x	FFFF ₁₆	←
operand[26]			
operand[27]	image_size_y	FFFF ₁₆	←
operand[28]			
operand[29]	next_pic	FFFF ₁₆	←
operand[30]			
operand[31]	next_page	FFFF ₁₆	←
operand[32]			

For the failed CAPTURE STATUS command with the *subfunction* of *get_status*, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.61 – Possible result value for the rejected get_status subfunction

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified print_job_ID does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.
Other values	-	Reserved

6.2.3.2.3.2 Get the supported values for the specified print content format

The query *subfunction* for the CAPTURE STATUS command is used to retrieve the maximum *data_size*, *image_size_x*, and *image_size_y* for the specified *print_job_ID* and *image_format_specifier*.

The following Table 6.62 shows the sample command-response for a successful CAPTURE STATUS command with the *subfunction* of *query*.

Table 6.62 – Successful command-response for the CAPTURE STATUS command with the subfunction of query

	opcode, operands	STATUS command	STABLE response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	80 ₁₆ (query)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	00 ₁₆ (no error)
operand[3]	destination_plug	FF ₁₆	←
operand[4]	-	FF ₁₆	00 ₁₆
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:		<i>record ID</i>	←
operand[16]	image_format_specifier	image_format	←
operand[17]			
operand[18]	-	FFFF ₁₆	0000 ₁₆
operand[19]			
operand[20]	data_size	FFFFFFFF ₁₆	<i>maximum data_size for the specified image format</i>
:			
operand[24]			
operand[25]	image_size_x	FFFF ₁₆	<i>maximum image_size_x for the specified image format</i>
operand[26]			
operand[27]	image_size_y	FFFF ₁₆	<i>maximum image_size_y for the specified image format</i>
operand[28]			
operand[29]	next_pic	FFFF ₁₆	←
operand[30]	next_page	FFFF ₁₆	←
operand[31]			
operand[32]			

If the *image_format_specifier* had been FE00₁₆ (Unit plug defined) or FE01₁₆ (Don't care), no information will be returned and the following fields of the response frame remain unchanged from the STATUS command frame:

- data_size = FFFFFFFF₁₆
- image_size_x = FFFF₁₆
- image_size_y = FFFF₁₆

If the specified *image_format_specifier* had been supported by the target and the target could not decide what values to be returned, the target sets *data_size*, *image_size_x*, and *image_size_y*, as follows:

- data_size = FFFFFFFF₁₆
- image_size_x = FFFF₁₆
- image_size_y = FFFF₁₆

The following Table 6.63 shows the sample command-response for a failed CAPTURE STATUS command with the *subfunction* of query.

Table 6.63 – Failed command-response for the CAPTURE STATUS command with the subfunction of query

	opcode, operands	STATUS command	REJECTED response
opcode	CAPTURE	42 ₁₆	←
operand[0]	subfunction	80 ₁₆ (query)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result value</i>
operand[3]	destination_plug	FF ₁₆	←
operand[4]	-	FF ₁₆	←
operand[5]	print_job_ID	FFFF ₁₆	←
:			
operand[12]		<i>record ID</i>	←
operand[13]			
:			
operand[16]	image_format_specifier	FFFF ₁₆	←
operand[17]		FFFF ₁₆	←
operand[18]			
operand[19]	-	FFFF ₁₆	←
operand[20]	data_size	FFFFFFFF ₁₆	←
operand[21]			
:			
operand[24]	image_size_x	FFFF ₁₆	←
operand[25]			
operand[26]	image_size_y	FFFF ₁₆	←
operand[27]			
operand[28]	next_pic	FFFF ₁₆	←
operand[29]			
operand[30]	next_page	FFFF ₁₆	←
operand[31]			
operand[32]			

For the failed CAPTURE STATUS command with the *subfunction* of query, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.64 – Possible result value for the rejected query subfunction

Value	Symbol	Meaning
80 ₁₆	format_error	The command execution had been aborted, because the image format is not supported or unacceptable now.
86 ₁₆	no_job_exists	The print job with specified print_job_ID does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.
Other values	-	Reserved

6.2.4 CAPTURE REF command

CAPTURE REF command is used to retrieve a markup language document ,e.g. XHTML-Print , for the specified job, and get information about capabilities of the printer subunit for a markup language type . If a markup language data refers image data in itself, the printer subunit acts as a controller for the data source subunit , e.g. camera storage subunit , inside of data source unit and requests to retrieve the image data from the data source subunit.

CAPTURE REF CONTROL command shall be issued only one time per one job .

6.2.4.1 CAPTURE REF CONTROL command

CAPTURE REF CONTROL command (ctype = CONTROL) is used to receive an XHTML / HTML document(s) from the asynchronous plug, , or from an /external plug, or a source plug of other subunits inside the AV unit. The format of CAPTURE REF CONTROL command is illustrated by the figure below.

	length	msb						lsb	
opcode	1		CAPTURE REF(44 ₁₆)						
operand[0]	1		subfunction						
operand[1]	1		status						
operand[2]	1		result						
operand[3]	1		destination_plug						
operand[4]	1		source_plug						
operand[5]	12		print_job_ID						
:									
operand[16]									
operand[17]	2		content_format_specifier						
operand[18]									
operand[19]	2		reserved						
operand[20]									
operand[21]	2		producer_node_ID						
operand[22]									
operand[23]	1		subunit_type:5bit / subunit_ID:3bit						
operand[24]	1		total_path_length						
operand[25]	1		base_path_length						
operand[26]	255		absolute_file_path						
:									
operand[280]									

Figure 6.15 – CAPTURE REF CONTROL command basic format

6.2.4.1.1 Field definitions

subfunction: This field indicates the operation to be taken by the target as defined in the Table 5.65 below.

Table 6.65 – subfunction field definitions for CAPTURE REF CONTROL command

Value	Symbol	Meaning
01 ₁₆	retrieve	retrieve a markup language document
02 ₁₆	abort	Abort retrieving a markup language document
04 ₁₆	resume	resume retrieving a markup language document after a serial bus reset
FF ₁₆	get_status	Get the operation status (for STATUS command)
Other values	-	Reserved

status: In case of the response frame of the CONTROL REF command or the STATUS command, this field indicates the state of the target, as defined in the Table 6..66. In case of the CONTROL command frame or STATUS command frame, this field shall be set to FF₁₆.

Table 6.66 – status field definitions

Value	Symbol	Meaning
00 ₁₆	idle	The command execution had been finished successfully, or no command is under execution.
01 ₁₆	retrieving_processing	The target is now retrieving the data and processing it.
02 ₁₆	resume_wait	The serial bus reset occurred, the target is waiting to be resumed.
FF ₁₆	-	No meaning
Other values	-	Reserved

result: This field indicates the result of the command execution in case of the response frame of the CONTROL REFcommand, as defined in the Table 6.67. In case of the CONTROL command frame or STATUS command frame, this field shall be set to FF₁₆.

Table 6.67 – result field definitions

Value	Symbol	Meaning
00 ₁₆	no_error	The command execution had been finished successfully, or no command is under execution.
80 ₁₆	format_error	The command execution had been aborted, because the content format is not supported or unacceptable.
83 ₁₆	invalid_format	The command execution had been aborted, because the content format is not the same as specified by the command frame
86 ₁₆	no_job_exists	The print job with specified <i>print_job_ID</i> does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy
88 ₁₆	no_process_to_abort	The command is rejected because currently there is no capturing process undergoing.
90 ₁₆	no_pending_process	The resume subfunction is rejected, because there is no pending capturing process. The capturing may be finished.
91 ₁₆	aborted	The command is aborted because the controller issued the CAPTURE REF CONTROL command with the <i>subfunction</i> of abort.
92 ₁₆	queue_full	The command is rejected because the job queue is full.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred
FF ₁₆	-	No meaning
Other values	-	Reserved

destination_plug: This field indicates the destination plug number (00₁₆ - 1E₁₆) of this subunit to input the print content. If the specified plug number is not supported by the printer subunit, it returns NOT IMPLEMENTED response.

source_plug: This field indicates the source plug number (00₁₆ - 1E₁₆) of data source subunit , e.g. a camera storage subunit, to output the print content..

print_job_ID: This field specifies the print job entry, as defined in the Table 6.5.

content_format_specifier: This field specifies the format of the retrieving print content. as defined in the table 6.68. The upper byte of this field indicates the category of the specified print content format. The lower byte of this field specifies the detailed print content format.

Table 6.65 – content_format_specifier definitions

Value(MSB)	Value(LSB)	Meaning
10₁₆	-	<i>XML File formats category</i>
10 ₁₆	00 ₁₆	XHTML-Print 1.0 (CSS Print)
10 ₁₆	01 ₁₆	XHTML-Print 1.0 (CSS Print Enhanced)
20₁₆	-	<i>HTML File formats category</i>
20 ₁₆	00 ₁₆	HTML3.2
20 ₁₆	01 ₁₆	HTML4.0 Transitional
20 ₁₆	02 ₁₆	HTML4.0 Strict
20 ₁₆	03 ₁₆	HTML4.01 Transitional
20 ₁₆	04 ₁₆	HTML4.01 Strict
80₁₆ – 8F₁₆	-	<i>Vendor Dependent Formats category</i>
80 ₁₆ – 8F ₁₆	00 ₁₆ – FF ₁₆	Vendor Dependent Formats
FE ₁₆	00 ₁₆	Unit plug defined
FE ₁₆	01 ₁₆	Don't care
Other values		Reserved

NOTE The value of FE01₁₆ (Don't care) is used when the data will be retrieved from other subunit inside the same unit.

producer_node_ID: This field indicates the node_ID of unit which acts as producer and has a data source subunit , e.g. a camera storage subunit.

subunit_type: This field indicates the type of a data source subunit , e.g. a camera storage subunit.

subunit_ID: This field indicates subunit ID of a data source subunit , e.g. a camera storage subunit.

total_path_length: This field indicates meaningful lengths bytes of **absolute_file_path** fields .The maximum value of this field is FF₁₆.

base_path_length: This field indicates meaningful lengths bytes of directory part of **absolute_file_path**. The maximum value of this field is FF₁₆.

absolute_file_path: This field indicates absolute file path to print content . URI notation is allowed if print content places on Internet.

For example;

absolute_file_path : C:/print_data/1234

base_path_length: 15

total_path_length: 19

The absolute path is used to request retrieving for a data source subunit.

6.2.4.1.2 CAPTURE REF CONTROL command operations

This section describes how the printer subunit behaves by the CAPTURE REF CONTROL commands.

6.2.4.1.2.1 CAPTURE REF CONTROL command state machine

The following shows the state transitions of the normal CAPTURE REF command operation. Note that this state machine only describes the state transitions triggered by the CAPTURE REF CONTROL commands or the serial bus reset or other rejection, and there is no description about other events like STATUS command or invalid command rejection.

The state transitions of the CAPTURE REF CONTROL command operation is same as CAPTURE CONTROL command operation fundamentally.

The difference between CAPTURE CONTROL command operation and is CAPTURE REF CONTROL command operation below;

In CAPTURE CONTROL command , *subfunction* of receive is used to start capturing.On the contrary ,in CAPTURE REF CONTROL command , *subfunction* of reterieve is used to start capturing,

The state transitions relative to CAPTURE CONTROL command with *subfunction* of receive , substitutes for one of CAPTURE CONTROL command with *subfunction* of receive.

CAPTURE REF operation state machine states are described below.

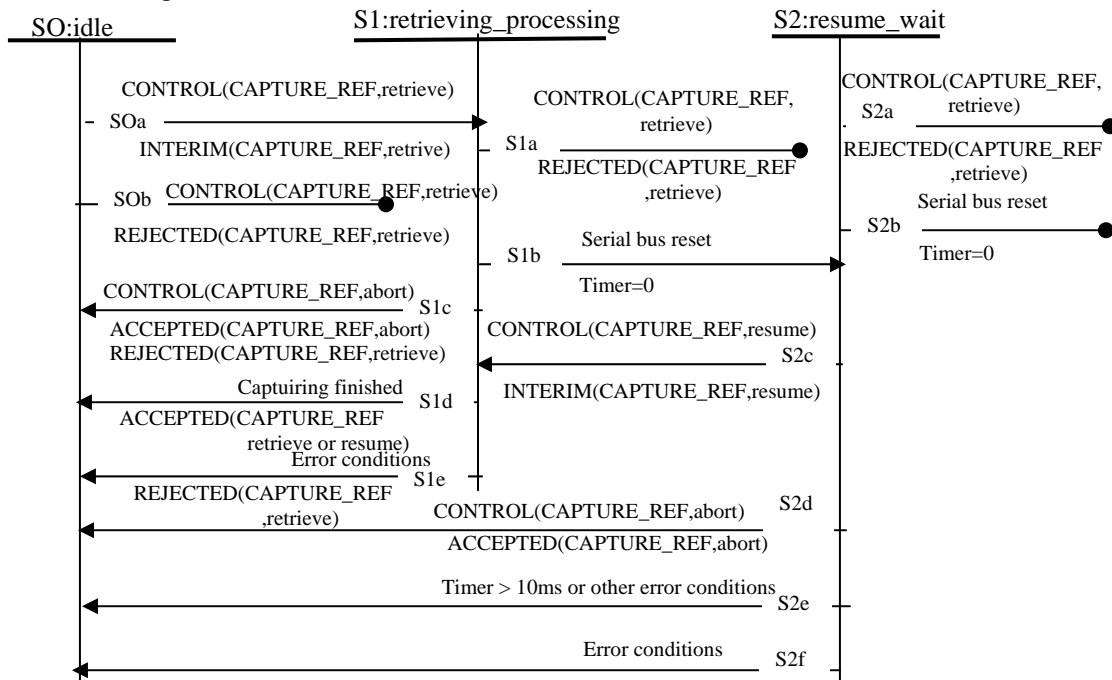


Figure 6-2 – CAPTURE REF CONTROL command operation state machine

S0: idle, The CAPTURE REF CONTROL command execution had been finished, or no CAPTURE REF CONTROL command is under execution.

S1: retrieving_processing, The target is now retrieving the data and / or processing it.

S2: resume_wait, The serial bus reset occurred, the target is waiting to be resumed.

6.2.4.1.2.1.1 State machine transitions

S0a. When the target receives the valid CAPTURE REF CONTROL command with the *subfunction* of retrieve, it returns the INTERIM response to the controller and transitions to the state S1.

S0b. When the target receives the invalid CAPTURE REF CONTROL command with the *subfunction* of retrieves or could not accept the command, it returns the REJECTED response to the controller.

S1a. When the target receives the additional CAPTURE REF CONTROL command with the *subfunction* of retrieves, it returns the REJECTED response to the controller, indicating the subunit is busy.

S1b. When the serial bus reset occurred, the target starts the timer to detect the resumption timeout, then transitions to the state S2.

S1c. When the target receives the valid CAPTURE REF CONTROL command with the *subfunction* of abort, it returns the ACCEPTED response, then returns REJECTED response for the previously issued CAPTURE CONTROL command with the *subfunction* of retrieve or resume, then transitions to the state S0.

S1d. When the target finished capturing process, it returns ACCEPTED response to the controller.

S1e. When the target detects some error conditions and stops capturing process, it returns the REJECTED response to the controller, then transitions to the state S0.

S2a. When the target receives the additional CAPTURE REF CONTROL command with the *subfunction* of retrieves, it returns the REJECTED response to the controller, indicating the subunit is busy.

S2b. When the serial bus reset occurred, the target restarts the timer to detect the resumption timeout.

S2c. When the target receives the valid CAPTURE REF CONTROL command with the *subfunction* of resume, it returns the ACCEPTED response, then transitions to the state S1.

S2d. When the target receives the valid CAPTURE REF CONTROL command with the *subfunction* of abort, it returns the ACCEPTED response, then transitions to the state S0.

S2d. When the target finished capturing process, it returns ACCEPTED response to the controller.

S2e. After the target detects the resumption timeout (10 seconds), it may stop capturing process, then transitions to the state S0.

S2f. When the target detects some error conditions and stops capturing process, transitions to the state S0.

6.2.4.1.2.2 Command resumption from serial bus reset

This section defines the behavior of the controller and target when a serial bus reset occurred. There are some scenarios for the bus-reset recovery.

(1) No influence by the serial bus reset (Internal connections, External plug connection)

When the subunit had been executing the capturing from the subunit inside the unit, or from the external plug, the serial bus reset only cancels the AV/C command response transactions, and the capturing process will be continued by the subunit. The controller should issue the resume subfunction to restart the AV/C command transaction.

(2) Asynchronous connections

When the bus reset occurred, the asynchronous connection transport stops its activities until the controller restores the connection. The controller shall restore the asynchronous connection first, then issue the CAPTURE REF resume subfunction to restart the AV/C command transaction.

(4) Print content source disappeared

After the serial bus reset, the print content source unit might be disappeared from the bus. In this case, the controller could not restore the connection, so the controller should abort the CAPTURE REF command by abort subfunction.

6.2.4.1.2.2.1 Target behavior

The AV/C command transaction of CAPTURE REF CONTROL command is aborted by the serial bus reset. But the target is expected to continue the data retrieving and processing if possible.

Until receipt of the CAPTURE REF CONTROL command with the subfunction of resume or 10 seconds timeout detected, the target should hold the previously issued CAPTURE REF CONTROL command information to prepare the resumption.

It is the controller's responsibility to restore the data transport path prior to the CAPTURE REF CONTROL command with the *subfunction* of resume. But if there is no data transport path restored when the target receives the CAPTURE REF CONTROL command with the *subfunction* of resume, the target rejects the resumption command.

6.2.4.1.2.2.2 Controller behavior

After the serial bus reset has been detected, the controller is expected to restore the data transport path (e.g. Asynchronous connection) first, then issue the CAPTURE REF CONTROL command which the *subfunction* of resume. As the target detects the resumption timeout after 10 seconds had passed from the latest serial bus reset, the controller should issue the resumption command within 10 seconds to the target. Even if the controller tried to restore the CAPTURE REF process, the CAPTURE REF operation may fail

The controller also may abort the capturing operation by CAPTURE REF CONTROL command with the *subfunction* of abort.

6.2.4.1.2.2.3 The final response loss

The final response of CAPTURE REF CONTROL command with the *subfunction* of retrieve may be lost by the serial bus reset. If the controller attempts to restore the capturing process which had already finished, the CAPTURE CONTROL command with the *subfunction* of resume, is rejected with the *result* field value of no_pending_process.

The controller can check the status of the printer subunit by CAPTURE REF STATUS command with the *subfunction* value set to FF₁₆. The *status* field include the current capturing status of the target.

6.2.4.1.2.3 Executing the capturing operation

The CAPTURE REF CONTROL command with the *subfunction* of retrieve , directs the print job to retrieve the print content. The target returns ACCEPTED response for the CAPTURE REF CONTROL command with the *subfunction* of retrieve to the controller when the print content had been captured successfully. (If the target could not finish the capturing operation, it returns REJECTED response.)

The controller can expect the target is ready to process the next CAPTURE REF CONTROL command with the *subfunction* of retrieve.

The following Table 6.66 shows the sample command-response for a successful CAPTURE REF CONTROL command.

Table 6.66 – Successful command-response for the CAPTURE REF CONTROL command with the subfunction of retrieve

	opcode, operands	CONTROL command	ACCEPTED response
opcode	CAPTURE REF	44 ₁₆	←
operand[0]	subfunction	01 ₁₆ (retrieve)	←
operand[1]	status	FF ₁₆	00 ₁₆ (idle)
operand[2]	result	FF ₁₆	00 ₁₆ (no error)
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	source_plug	<i>source_plug number</i>	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:		<i>record ID</i>	←
operand[12]			
operand[13]			
:	content_format_specifier	<i>content format</i>	←
operand[16]			
operand[17]	reserved	0000 ₁₆	←
operand[18]			
operand[19]	producer_node_ID	<i>node_ID</i>	←
operand[20]			
operand[21]	subunit_type / subunit_ID	<i>subunit_type / subunit_ID</i>	←
operand[22]			
operand[23]	total_path_length	<i>length (bytes) of absolute_file_path</i>	
operand[24]	base_path_length	<i>length (bytes) of directory part of absolute_file_path</i>	←
operand[25]	absolute_file_path	<i>Absolute file path to print content</i>	←
operand[26]			
operand[280]			

The following Table 6.67 shows the sample command-response transaction for a failed CAPTURE REF CONTROL command.

Table 6.67 – Failed command-response for CAPTURE REF CONTROL command with the subfunction of retrieve

	opcode, operands	CONTROL command	REJECTED response
opcode	CAPTURE REF	44 ₁₆	←
operand[0]	subfunction	01 ₁₆ (retrieve)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result_value</i>
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	source_plug	<i>source_plug number</i>	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:		<i>record ID</i>	←
operand[12]			
operand[13]			
:	content_format_specifier	<i>content format</i>	←
operand[16]			
operand[17]	reserved	0000 ₁₆	←
operand[18]			
operand[19]	producer_node_ID	<i>Node_ID</i>	←
operand[20]			
operand[21]	subunit_type / subunit_ID	<i>subunit_type / subunit_ID</i>	←
operand[22]			
operand[23]	total_path_length	<i>length (bytes) of absolute_file_path</i>	←
operand[24]	base_path_length	<i>length (bytes) of directory part of absolute_file_path</i>	←
operand[25]	absolute_file_path	<i>Absolute file path to print content</i>	←
operand[26]			
operand[280]			

NOTE NOTE When the target could not accept the command because the *content_format_specifier* or *data_size* or *file_name* or *base_path* values had not been supported, it returns REJECTED response with result.

If other field of the CONTROL command frame had not been supported by the target, it returns NOT IMPLEMENTED response.

For the failed CAPTURE REF CONTROL command with the *subfunction* of retrieve , the *result* field of the REJECTED response field contains the result value to indicate why the command had been rejected, as follows.

Table 6.68 – Possible result value for the rejected receive subfunction

Value	Symbol	Meaning
80 ₁₆	format_error	The command execution had been aborted, because the print content format is not supported or unacceptable now.
83 ₁₆	invalid_format	The command execution had been aborted, because the print content format is not the same as specified by the command frame
86 ₁₆	No_job_exists	The print job with specified print_job_ID does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy
8C ₁₆	already_issued	The CAPTURE REF CONTROL command has been already issued for the job
91 ₁₆	aborted	The command is aborted because the controller issued the CAPTURE REF CONTROL command with the subfunction of abort.
92 ₁₆	queue_full	The command is rejected because the job queue is full.
FE ₁₆	Any_other_error	The command execution had been aborted, because any other error occurred
Other values	-	Reserved

6.2.4.1.2.4 Aborting the capturing operation after the serial bus reset

The controller can abort the undergoing capturing process by CAPTURE REF CONTROL command with the *subfunction* of abort.

The CONTROL command frame shall be the same as previously issued CAPTURE REF CONTROL command with the *subfunction* of retrieve , except for the subfunction field value.

If the CAPTURE REF CONTROL command with the *subfunction* of abort had been accepted by the target, it returns ACCEPTED response, and the target starts to stop the print content data capturing operation. When the target had “almost” finished the previously issued CAPTURE REF CONTROL command with the subfunction of retrieve , the target may return REJECTED response for the abort *subfunction*, with the *result* of busy.

After the target returned the ACCEPTED response for the CAPTURE REF CONTROL command with the *subfunction* of abort, the target returns REJECTED response

After the target directs the print job to retrieve the print content data. , the target returns ACCEPTED response for the CAPTURE REF CONTROL command with the *subfunction* of resume to the controller when the print content data had been captured successfully.

The following Table 6.69 shows the sample command-response for a successful CAPTURE REF CONTROL command with the subfunction of abort.

Table 6.69 – Successful command-response for the CAPTURE REF CONTROL command with the subfunction of abort

	opcode, operands	CONTROL command	ACCEPTED response
opcode	CAPTURE REF	44 ₁₆	←
operand[0]	subfunction	02 ₁₆ (abort)	←
operand[1]	status	FF ₁₆	00 ₁₆ (idle)
operand[2]	result	FF ₁₆	00 ₁₆ (no error)
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	source_plug	<i>source_plug number</i>	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:		<i>record ID</i>	←
operand[12]			
operand[13]			
:	content_format_specifier	<i>content format</i>	←
operand[16]			
operand[17]	reserved	0000 ₁₆	←
operand[18]			
operand[19]	producer_node_ID	<i>node_ID</i>	←
operand[20]			
operand[21]	subunit_type / subunit_ID	<i>subunit_type / subunit_ID</i>	←
operand[22]			
operand[23]	total_path_length	<i>length (bytes) of absolute_file_path</i>	←
operand[24]	base_path_length	<i>length (bytes) of directory part of absolute_file_path</i>	←
operand[25]	absolute_file_path	<i>Absolute file path to print content</i>	←
operand[26]			
:			
operand[280]			
operand[409]			

The following Table 6.70 shows the sample command-response transaction for a failed CAPTURE REF CONTROL command with the subfunction of abort.

Table 6.70 – Failed command-response for CAPTURE REF CONTROL command with the subfunction of abort

	opcode, operands	CONTROL command	REJECTED response
opcode	CAPTURE REF	44 ₁₆	←
operand[0]	subfunction	02 ₁₆ (abort)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result_value</i>
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	source_plug	<i>source_plug number</i>	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:		<i>record ID</i>	←
operand[12]			
operand[13]			
:	content_format_specifier	<i>content format</i>	←
operand[16]			
operand[17]	reserved	0000 ₁₆	←
operand[18]			
operand[19]	producer_node_ID	<i>node_ID</i>	←
operand[20]			
operand[21]	subunit_type / subunit_ID	<i>subunit_type / subunit_ID</i>	←
operand[22]			
operand[23]	total_path_length	<i>length (bytes) of absolute_file_path</i>	←
operand[24]	base_path_length	<i>length (bytes)</i>	←
operand[25]	base_path	<i>path to print content directorypath to a markup language data</i>	←
operand[26]			
operand[280]			

NOTE When the target could not accept the command because the *content_format_specifier* or *data_size* or *file_name* or *base_path* values had not been the same as the previously issued CONTROL command frame, it returns REJECTED response with the result of no_process_to_abort. If the specified *print_job_ID* is not supported by the target, it returns REJECTED response with the result of no_job_exists. If other field of the CONTROL command frame had not been supported by the target, it returns NOT IMPLEMENTED response.

For the failed CAPTURE REF CONTROL command with the subfunction of abort, the result field of the REJECTED response frame contains the *result* value to indicate why the command had been rejected, as follows.

Table 6.71 – Possible result value for the rejected abort subfunction

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified print_job_ID does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy
88 ₁₆	no_process_to_abort	The command is rejected because currently there is no capturing process undergoing.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred
Other values	-	Reserved

6.2.4.1.2.5 Restoring the capturing operation after the serial bus reset

The command response transaction of the CAPTURE REF CONTROL command with the *subfunction* of retrieve , is aborted by the serial bus reset, but the target may continue to process the data.

When the controller requests to restore the command response transaction, it issues the CAPTURE REF CONTROL command with the *subfunction* of resume. The CONTROL command frame shall be the same as previously issued CAPTURE REF CONTROL command with the *subfunction* of retrieve (or resume), except for the subfunction field value and producer_node_ID field value .Then the controller shall search new node_ID of data source subunit and set it to producer_node_ID field .

After the target directs the print job to receive the print content the target returns ACCEPTED response for the CAPTURE REF CONTROL command with the *subfunction* of resume to the controller when the print content had been captured successfully.

The following Table 6.72 shows the sample command-response for a successful CAPTURE REF CONTROL command with the subfunction of resume.

Table 6.72 – Successful command-response for the CAPTURE REF CONTROL command with the subfunction of resume

	opcode, operands	CONTROL command	ACCEPTED response
opcode	CAPTURE REF	44 ₁₆	←
operand[0]	subfunction	04 ₁₆ (resume)	←
operand[1]	status	FF ₁₆	00 ₁₆ (idle)
operand[2]	result	FF ₁₆	00 ₁₆ (no error)
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	source_plug	<i>source_plug number</i>	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:			
operand[12]			
operand[13]			
:	<i>record ID</i>	←	
operand[16]			
operand[17]	content_format_specifier	<i>content format</i>	←
operand[18]			
operand[19]	reserved	0000 ₁₆	←
operand[20]	producer_node_ID	<i>node_ID</i>	←
operand[21]			
operand[22]	subunit_type / subunit_ID	<i>subunit_type / subunit_ID</i>	←
operand[23]			
operand[24]	total_path_length	<i>length (bytes) of absolute_file_path</i>	←
operand[25]	base_path_length	<i>length (bytes) of directory part of absolute_file_path</i>	←
operand[26]	absolute_file_path	<i>Absolute file path to print content</i>	←
:			
operand[280]			

The following Table 6.73 shows the sample command-response transaction for a failed CAPTURE REF CONTROL command with the subfunction of resume.

Table 6.73 – Failed command-response for CAPTURE REF CONTROL command with the subfunction of resume

	opcode, operands	CONTROL command	REJECTED response
opcode	CAPTURE REF	44 ₁₆	←
operand[0]	subfunction	04 ₁₆ (resume)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	<i>result_value</i>
operand[3]	destination_plug	<i>destination_plug number</i>	←
operand[4]	source_plug	<i>source_plug number</i>	←
operand[5]	print_job_ID	<i>controller EUI-64</i>	←
:		<i>record ID</i>	←
operand[12]			
operand[13]			
:	content_format_specifier	<i>content format</i>	←
operand[16]			
operand[17]	reserved	0000 ₁₆	←
operand[18]			
operand[19]	producer_node_ID	<i>node_ID</i>	←
operand[20]			
operand[21]	subunit_type / subunit_ID	<i>subunit_type / subunit_ID</i>	←
operand[22]			
operand[23]	total_path_length	<i>length (bytes) of absolute_file_path</i>	←
operand[24]	base_path_length	<i>length (bytes) of directory part of absolute_file_path</i>	←
operand[25]	absolute_file_path	<i>Absolute file path to print content</i>	←
operand[26]			
:			
operand[280]			
operand[409]			

NOTE When the target could not accept the command because the *content_format_specifier* or *data_size* or *file_name* or *base_path* values had not been supported, it returns REJECTED response with the result of *no_pending_process*. If the specified *print_job_ID* is not supported by the target, it returns REJECTED response with the *result* of *no_job_exists*.

If other field of the CONTROL command frame had not been supported by the target, it returns NOT IMPLEMENTED response.

For the failed CAPTURE REF CONTROL command with the subfunction of resume, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.74 – Possible result value for the rejected resume subfunction

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified print_job_ID does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy
8B ₁₆	invalid_node_ID	The node ID is rejected because it is invalid
90 ₁₆	no_pending_process	The resume subfunction is rejected, because there is no pending capturing process. The capturing may be finished.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred
Other values	-	Reserved

6.2.4.2 CAPTURE REF STATUS command

The CAPTURE REF STATUS command (ctype=STATUS) is used to retrieve the information about the capturing operations.

6.2.4.2.1 Get the current capturing status

The *get_status subfunction* for the CAPTURE REF STATUS command is used to retrieve the current capture operation status.

The following Figure 6-17 indicates the command frame format of the CAPTURE STATUS command with the *subfunction* of *get_status*.

	length	msb						lsb
opcode	1	CAPTURE REF(44 ₁₆)						
operand[0]	1	subfunction						
operand[1]	1	FF ₁₆						
operand[2]	1	FF ₁₆						
operand[3]	1	FF ₁₆						
operand[4]	1	FF ₁₆						
operand[5]	12	print_job_ID						
:								
operand[16]	2	content_format_specifier						
operand[17]								
operand[18]	2	FFFF ₁₆						
operand[19]								
operand[20]	2	FFFF ₁₆						
operand[21]								
operand[22]	1	FF ₁₆						
operand[23]								
operand[24]	1	FF ₁₆						
operand[25]								
operand[26]	255	FF... FF ₁₆						
:								
operand[280]								

Figure 6-17 – CAPTURE REF STATUS command frame with subfunction get_status

The following Figure 6-18 indicates the response frame format of the CAPTURE REF STATUS command with the *subfunction* of get_status.

	length	msb						lsb
opcode	1	CAPTURE REF(44 ₁₆)						
operand[0]	1	subfunction						
operand[1]	1	status						
operand[2]	1	result						
operand[3]	1	destination_plug						
operand[4]	1	source_plug						
operand[5]	12	print_job_ID						
:								
operand[16]								
operand[17]	2	content_format_specifier						
operand[18]								
operand[19]	2	FFFF ₁₆						
operand[20]								
operand[21]	2	producer_node_ID						
operand[22]								
operand[23]	1	subunit_type:5bit / subunit_ID:3bit						
operand[24]	1	total_path_length						
operand[25]	1	base_path_length						
operand[26]	255	absolute_file_path						
:								
operand[280]								

Figure 6-18 – CAPTURE REF status response frame with subfunction get_status

The detail of the CAPTURE REF status command operation is described in “6.2.4.2.3 CAPTURE REF STATUS command operations”.

6.2.4.2.1.1 Field definitions

subfunction: This field indicates the operation to be taken by the target as defined in the Table 6.75 below.

Table 6.75 – subfunction field definitions for CAPTURE REF STATUS command

Value	Symbol	Meaning
80 ₁₆	query	Get the support status for the specified content_format_specifier (for STATUS command)
FF ₁₆	get_status	Get the operation status (for STATUS command)
Other values	-	Reserved

print_job_ID: When a subfunction field value is query, this field value shall be FFFFFFFFFFFFFFFFFFFFFFFFFF₁₆ for both command and response frame. When a subfunction field value is get_status, this field value shall specify the print job in the queue for command frame.

content_format_specifier: When a subfunction field value is query, this field value shall specify the print content format as defined in “Table 6.65 – content_format_specifier definitions” for both command and response frame. When the *subfunction* field value is get_status, this field value shall be FFFF₁₆ for command frame.

Other fields are defined according to the *subfunction* value, as illustrated below.

Table 6.76 – Definitions of other fields

Field	subfunction	Meaning
status	query	No meaning (FF ₁₆)
	get_status	Current status of capturing (see 6.2.3.1.2.1.1)
result	query	Command execution result
	get_status	

6.2.4.2.2 Get format support level

The query subfunction for the CAPTURE REF STATUS command is used to check whether the printer subunit support specified print content format or not.

The following Figure 6-19 indicates the command frame format of the CAPTURE REF STATUS command with the *subfunction* of query.

	length	msb						lsb
opcode	1	CAPTURE REF(44 ₁₆)						
operand[0]	1	80 ₁₆ (query)						
operand[1]	1	FF ₁₆						
operand[2]	1	FF ₁₆						
operand[3]	1	FF ₁₆						
operand[4]	1	FF ₁₆						
operand[5]	12	FFFFFFFFFFFFFFFF ₁₆						
:								
operand[16]								
operand[17]	2	content_format_specifier						
operand[18]								
operand[19]	2	FFFF ₁₆						
operand[20]								
operand[21]	2	FFFF ₁₆						
operand[22]								
operand[23]	1	FF ₁₆						
operand[24]	1	FF ₁₆						
operand[25]	1	FF ₁₆						
operand[26]	255	FF... FF ₁₆						
:								
operand[280]								

Figure 6-19 – CAPTURE REF STATUS command frame with subfunction query

The following Figure 6-20 indicates the response frame format of the CAPTURE REF STATUS command with the *subfunction* of query.

The CAPTURE REF STATUS command with the *subfunction* of query can be used whenever the specified print job is opened (the status of the specified print job is **S1:opened_idle** or **S2:opened_capturing**).

The detail of the CAPTURE status command operation is described in “6.2.4.2.3 CAPTURE REF STATUS command operations”

Figure 6-20 – CAPTURE REF status response frame with subfunction query

If the printer subunit supports markup language data specified by *content_format_specifier* , *no_error*

	length	msb						lsb
opcode	1	CAPTURE REF(44 ₁₆)						
operand[0]	1	80 ₁₆ (query)						
operand[1]	1	FF ₁₆						
operand[2]	1	result_value						
operand[3]	1	FF ₁₆						
operand[4]	1	FF ₁₆						
operand[5]	12	FFFFFFFFFFFFFFFF ₁₆						
:								
operand[16]	2	content_format_specifier						
operand[17]								
operand[18]	2	FFFF ₁₆						
operand[19]								
operand[20]	2	FFFF ₁₆						
operand[21]								
operand[22]	1	FF ₁₆						
operand[23]								
operand[24]	1	FF ₁₆						
operand[25]								
operand[26]	1	FF ₁₆						
:								
operand[280]	255	FF... FF ₁₆						

(00₁₆) shall be set to result_value field . And if not supports , format_error(80₁₆) shall be set . .

6.2.4.2.3 CAPTURE REF STATUS command operations

There are two *subfunction* values defined for the STATUS command operations.

6.2.4.2.3.1 Get the current capturing status

The *get_status subfunction* for the CAPTURE REF STATUS command is used to retrieve the current capture operation status.

The following Table 6.77 shows the sample command-response for a successful CAPTURE REF STATUS command with the *subfunction* of *get_status*.

Table 6.77 – Successful command-response for the CAPTURE REF STATUS command with the subfunction of *get_status*

	opcode, operands	STATUS command	STABLE response	
opcode	CAPTURE REF	44 ₁₆	←	
operand[0]	subfunction	FF ₁₆ (<i>get_status</i>)	←	
operand[1]	status	FF ₁₆	Current_status	
operand[2]	result	FF ₁₆	00 ₁₆ (no error)	
operand[3]	destination_plug	FF ₁₆	←	
operand[4]	source_plug	FF ₁₆	←	
operand[5]	print_job_ID	<i>controller</i> <i>EUI-64</i>	←	
:			<i>record ID</i>	←
operand[12]		content_format_specifier		FFFF ₁₆
operand[13]			reserved	FFFF ₁₆
:	producer_node_ID	FFFF ₁₆		←
operand[16]		subunit_type / subunit_ID	FFFF ₁₆	←
operand[17]	total_path_length		FF ₁₆	
operand[18]		base_path_length	FF ₁₆	←
operand[19]	absolute_file_path		FF....FF ₁₆	←
operand[20]				
operand[21]				
operand[22]				
operand[23]				
operand[24]				
operand[25]				
operand[26]				
:				
operand[280]				

In the response frame, “current status”, value is different according to the capturing status.

The following table indicates the parameter values that would be returned.

Table 6.78 – status response field values

Capturing state	current status
S0: idle	00₁₆
S1: retrieving_processing	01₁₆
S2: resume_wait	02₁₆

The following Table 6.79 shows the sample command-response for a failed CAPTURE REF STATUS command with the *subfunction* of *get_status*.

Table 6.79 – Failed command-response for the CAPTURE REF STATUS command with the subfunction of get_status

	opcode, operands	STATUS command	REJECTED response
opcode	CAPTURE REF	44 ₁₆	←
operand[0]	subfunction	FF ₁₆ (get_status)	←
operand[1]	status	FF ₁₆	←
operand[2]	result	FF ₁₆	result_value
operand[3]	destination_plug	FF ₁₆	←
operand[4]	source_plug	FF ₁₆	←
operand[5]	print_job_ID	controller EUI-64	←
:			
operand[12]			
operand[13]			
:	record ID	←	
operand[16]			
operand[17]	content_format_specifier	FFFF ₁₆	←
operand[18]			
operand[19]	reserved	FFFF ₁₆	←
operand[20]			
Operand[21]	producer_node_ID	FFFF ₁₆	←
Operand[22]			
Operand[23]	subunit_type / subunit_ID	FFFF ₁₆	←
operand[24]	total_path_length	FF ₁₆	←
operand[25]	base_path_length	FF ₁₆	←
operand[26]	absolute_file_path	FF.....FF ₁₆	←
:			
operand[280]			

For the failed CAPTURE REF STATUS command with the *subfunction* of get_status, the *result* field of the REJECTED response frame contains the result value to indicate why the command had been rejected, as follows.

Table 6.80 – Possible result value for the rejected get_status subfunction

Value	Symbol	Meaning
86 ₁₆	no_job_exists	The print job with specified print_job_ID does not exist.
87 ₁₆	busy	The command is rejected because the subunit is busy.
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.
Other values	-	Reserved

6.2.5 PRINTER STATUS command

The PRINTER STATUS command is used to get information of the status of the printer subunit.

6.2.5.1 PRINTER STATUS STATUS command

The PRINTER STATUS STATUS command (ctype=STATUS) is used to get information of the status of the printer subunit. The format of PRINTER STATUS STATUS command is illustrated by the figure below:

	length	msb						lsb
opcode	1	PRINTER STATUS (43 ₁₆)						
operand[0]	1	FF ₁₆						
operand[1]	1	FF ₁₆						
operand[2]	3	FF FF FF ₁₆						
:								
operand[4]								
operand[5]	12	FF FF FF FF FF FF FF FF FF FF FF FF ₁₆						
:								
operand[16]								

Figure 6-21 – PRINTER STATUS STATUS command frame

	length	msb						lsb
opcode	1	PRINTER STATUS (43 ₁₆)						
operand[0]	1	00 ₁₆						
operand[1]	1	line	status_category				message_number	
operand[2]	3	00 00 00 ₁₆						
:								
operand[4]								
operand[5]	12	print_job_ID						
:								
operand[16]								

Figure 6-22 – PRINTER STATUS STATUS command response frame

6.2.5.1.1 Field definitions

line: This bit indicates whether the printer subunit is “on-line” or “off-line”. If this bit is set to one, the printer subunit is “off-line”. When this bit is set to one (off-line), the printer engine can not receive any data from the queue. Some printer may have the button of “on-line” on its control panel to allow the user to pause the printer engine. Other printer which has no “on-line” button, may always set this bit to zero (on-line) in the STABLE response frame.

status_category: This field indicates the category of the error condition status, as defined in the table below.

Table 6.81 – status_category definitions

Value	Symbol	Meaning
0 ₁₆	no_error	The printer holds no error condition
1 ₁₆	just_wait	Possible to recover while the user is just waiting
2 ₁₆	request_to_recover	Possible to recover if the user does some recovery procedure to the printer
4 ₁₆	fatal_error	Fatal error occurred. There is no way to recover
Other values	-	Reserved

message_number: The value of *message_number* field, in combination with the value of *status_category* indicates the expected “User Interface” message about the subunit status. If the controller had the user interface which can be used to inform the error conditions of the printer subunit to the user, it is recommended to use the error messages as defined below.

Message_number values except the value of the following combinations are reserved.

Table 6.82 – Recommended error messages

status_category value	message_number value	Recommended message
0 ₁₆	0 ₁₆	(no recommended message)
1 ₁₆	0 ₁₆	“Just wait for a while.”
2 ₁₆	0 ₁₆	“Please check the printer.”
2 ₁₆	1 ₁₆	“Please check the paper.”
2 ₁₆	2 ₁₆	“Please check the ink.”
2 ₁₆	3 ₁₆	“Paper jammed.”
4 ₁₆	0 ₁₆	“The printer is broken.”

print_job_ID: This field returns the *print_job_ID* of the print job entry which *job_position* value is zero. If there is no print job being processed at the printer engine, this field is FFFFFFFFFFFFFFFFFFFFFFFF₁₆.

6.2.5.2 Command/response example

The following Table 6.83 shows the sample command-response for a successful PRINTER STATUS STATUS command.

Table 6.83 – Successful command-response for the PRINTER STATUS STATUS command

	opcode, operands	STATUS command	STABLE response
opcode	PRINTER STATUS	43 ₁₆	←
operand[0]	-	FF ₁₆	00 ₁₆
operand[1]	line, status_category, message_number	FF ₁₆	<i>current values</i>
operand[2]	-	FF FF FF ₁₆	000000 ₁₆
:			
operand[4]	print_job_ID	FF .. FF ₁₆	<i>current print_job_ID</i>
operand[5]			
:			
operand[16]			

The following Table 6.84 shows the sample command-response for a failed PRINTER STATUS STATUS command.

Table 6.84 – Failed command-response for the PRINTER STATUS STATUS command

	opcode, operands	STATUS command	REJECTED response
opcode	PRINTER STATUS	43 ₁₆	←
operand[0]	-	FF ₁₆	←
operand[1]	line, status_category, message_number	FF ₁₆	←
operand[2]	-	FF FF FF ₁₆	←
:			
operand[4]	print_job_ID	FF .. FF ₁₆	←
operand[5]			
:			
operand[16]			

As the target may return the REJECTED response when it can not return the information, the controller can simply retry the PRINTER STATUS STATUS command later.

6.2.6 VERSION command

The VERSION command is used to get the version information of the printer subunit.

6.2.6.1 VERSION STATUS command

The VERSION STATUS command (ctype = STATUS) is used to get the version information of the printer subunit. It can be used to get the latest specification version information, or to inquire the specific version number is supported or not. The format of VERSION STATUS command is illustrated in the Figure 6-23, and STABLE response is illustrated in the Figure 6-24 below.

	length	msb							lsb
opcode	1	VERSION (B0 ₁₆)							
operand[0]	1	FF ₁₆							
operand[1]	1	subunit_version_information							
operand[2]	31	FF .. FF ₁₆							
:									
operand[32]									

Figure 6-23 – VERSION STATUS command frame

	length	msb							lsb
opcode	1	VERSION (B0 ₁₆)							
operand[0]	1	FF ₁₆							
operand[1]	1	subunit_version_information							
operand[2]	31	subunit_version_information_dependent_field							
:									
operand[32]									

Figure 6-24 – VERSION STATUS command response frame

6.2.6.1.1 Field definitions

subunit_version_information: This field indicates the latest version number of the subunit specification it implements. The definition of this field depends on the each subunit specification. For this version of the AV/C Printer Subunit Specification, the value of this field is defined in the Table 6.85 as follows.

Table 6.85 – subunit_version_information field definitions

subunit_version_information	Meaning
10 ₁₆	Version 1.0 of the AV/C printer specification
20 ₁₆	Version 2.0 of the AV/C printer specification
Other values	Reserved

subunit_version_information_dependent_field: The definition of this field depends on the each subunit specification. For the AV/C Printer Subunit Specification 1.0 and later version, this field indicates the list of *implementation_profile_id* defined in the specification as follows.

operand	Length, bytes	Contents
02 ₁₆	1	implementation_profile_id [0]
03 ₁₆	1	implementation_profile_id [1]
:	:	:
31 ₁₆	1	implementation_profile_id [30]

Figure 6-25 – subunit_version_information_dependent_field definitions

The *implementation_profile_id* is specified by the version information field value. The value of *implementation_profile_id* is defined as follows:

Table 6.86 – implementation_profile_id field definitions

implementation_profile_id	Meaning
00 ₁₆	CAPTURE CONTROL commands with the image formats and sizes defined in Table 6.87 are accepted by the subunit.
01 ₁₆	CAPTURE REF CONTROL commands with the markup language data defined in Table 6.92 are accepted by the printer subunit.
FF ₁₆	No information
Other values	Reserved

Table 6.87 – Supported image formats and sizes

Image size	Image format			
	sRGB	YCC4:2:2 raw, chunky, progressive	YCC4:2:0 raw, chunky, progressive	Exif 2.1, compressed file
640 x 480	M	M	M	M
720 x 480	M	M	M	O
800 x 600	M	O	O	M
1024 x 768	M	O	O	M
1280 x 960	M	O	O	M
1280 x 720	M	M	M	O
1600 x 1200	O	O	O	O
1920 x 1080	M	M	M	O

NOTE M: Mandatory support, others (including the image formats and sizes that are not in the table) are optional. If the controller wants to use other image size or other image format, the controller can use CAPTURE STATUS command with the subfunction query to find out whether the desired parameter is supported by the subunit or not.

NOTE For the detail of the image format (e.g. Color space, Color parameters, ...), refer to “Annex D:”

NOTE The pixel aspect ratio may not be 1.0 x 1.0 in case of “YCC4:2:2” or “YCC4:2:0”. The printer subunit shall accept non- 1.0 x 1.0 YCC data. It is the printer subunit implementation dependent on how to handle the aspect ratio.

For a printer subunit defined by this specification returns the *subunit_version_information_dependent_field* as illustrated below.

operand	Length, bytes	Contents
02 ₁₆	1	00 ₁₆
03 ₁₆	1	FF ₁₆
:	:	:
32 ₁₆	1	FF ₁₆

Figure 6-26 – subunit_version_information_dependent_field values

Table 6.88 – Supported markup language type

markup language type	content_format_specifier value	Support level
XHTML-Print 1.0 (CSS Print)	1000 ₁₆	M
XHTML-Print 1.0 (CSS Print Enhanced)	1001 ₁₆	O
HTML 3.2	2000 ₁₆	O
HTML4.0 Transitional	2001 ₁₆	O
HTML 4.0 Strict	2002 ₁₆	O
HTML4.01 Transitional	2003 ₁₆	O
HTML 4.01 Strict	2004 ₁₆	O

NOTE M: Mandatory support, others (including markup language type that are not in the table) are optional. If the controller wants to use other markup language type, the controller can use CAPTURE REF STATUS command with the subfunction query to find out whether the desired type is supported by the subunit or not.

NOTE For the detail of the markup language type refer to References .

6.2.6.2 Get the latest version information

VERSION STATUS command can be used to get the latest version of the subunit specification. The format of the command frame is illustrated in the Figure 6-27 below, and the STABLE response frame is illustrated in the Figure 6-28.

	length	msb							lsb
opcode	1	VERSION (B0 ₁₆)							
operand[0]	1	FF ₁₆							
operand[1]	1	FF ₁₆							
operand[2]	31	FF .. FF ₁₆							
:									
operand[32]									

Figure 6-27 – VERSION STATUS command frame to get the latest version information

	length	msb							lsb
opcode	1	VERSION (B0 ₁₆)							
operand[0]	1	FF ₁₆							
operand[1]	1	subunit_version_information							
operand[2]	31	subunit_version_information_dependent_field							
:									
operand[32]									

Figure 6-28 – VERSION STATUS command response frame to get the latest version information

6.2.6.3 Get the support level of the specified version

VERSION STATUS command can also be used to get the support level of the specified version of the subunit specification. The format of the command frame is illustrated in the Figure 6-29 below, and the STABLE response frame is illustrated in the Figure 6-30.

	length	msb							lsb
opcode	1	VERSION (B0 ₁₆)							
operand[0]	1	FF ₁₆							
operand[1]	1	subunit_version_information							
operand[2]	31	FF .. FF ₁₆							
:									
operand[32]									

Figure 6-29 – VERSION STATUS command frame to get the support level of the specified version

	length	msb							lsb
opcode	1	VERSION (B0 ₁₆)							
operand[0]	1	FF ₁₆							
operand[1]	1	subunit_version_information							
operand[2]	31	subunit_version_information_dependent_field							
:									
operand[32]									

Figure 6-30 – VERSION STATUS command response frame to get the support level of the specified version

If a printer subunit is designed to support the specification specified by the *subunit_version_information* field value, it returns STABLE response with the *subunit_version_information_dependent_field* set to indicate its support information. The definitions of the *subunit_version_information* and *subunit_version_information_dependent_field* are the same as described in “6.2.6.2 Get the latest version information”.

If the subunit does not support the specification specified by the *subunit_version_information* field value, it returns NOT IMPLEMENTED response.

6.2.6.4 Command/response example

The following Table 6.89 shows the sample command-response for a successful VERSION STATUS command to get the latest version information.

Table 6.89 – Successful command-response for the VERSION STATUS command to get the latest version information

	opcode, operands	STATUS command	STABLE response
opcode	VERSION	B0 ₁₆	←
operand[0]	FF ₁₆	FF ₁₆	←
operand[1]	subunit_version_infor mation	FF ₁₆	20 ₁₆
operand[2]	subunit_version_infor mation_dependent_ field	FF ₁₆	01 ₁₆
:		FF .. FF ₁₆	←
operand[32]		FF ₁₆	←

The following Table 6.90 shows the sample command-response for a failed VERSION STATUS command to get the latest version information.

Table 6.90 – Failed command-response for the VERSION STATUS command to get the latest version information

	opcode, operands	STATUS command	REJECTED response
opcode	VERSION	B0 ₁₆	←
operand[0]	FF ₁₆	FF ₁₆	←
operand[1]	subunit_version_infor mation	FF ₁₆	←
operand[2]	subunit_version_infor mation_dependent_ field	FF ₁₆	←
:		FF .. FF ₁₆	←
operand[32]		FF ₁₆	←

As the target may return the REJECTED response when it can not return the information, the controller can simply retry the VERSION STATUS command later.

The following Table 6.91 shows the sample command-response for a successful VERSION STATUS command to get the support level of specified version.

Table 6.91 – Successful command-response for the VERSION STATUS command to get the support level of specified version

	opcode, operands	STATUS command	STABLE response
opcode	VERSION	B0 ₁₆	←
operand[0]	FF ₁₆	FF ₁₆	←
operand[1]	subunit_version_infor mation	10 ₁₆	←
operand[2]	subunit_version_infor mation_dependent_ field	FF ₁₆	00 ₁₆
:		FF .. FF ₁₆	←
operand[32]		FF ₁₆	←

The following Table 6.92 shows the sample command-response for a failed VERSION STATUS command to get the support level of specified version.

Table 6.92 – Failed command-response for the VERSION STATUS command to get the support level of specified version

	opcode, operands	STATUS command	NOT IMPLEMENTED response
opcode	VERSION	B0 ₁₆	←
operand[0]	FF ₁₆	FF ₁₆	←
operand[1]	subunit_version_information	10 ₁₆	←
operand[2]	subunit_version_information_dependent_	FF ₁₆	←
:	field	FF .. FF ₁₆	←
operand[32]		FF ₁₆	←

As the target may return the REJECTED response when it can not return the information, the controller can simply retry the VERSION STATUS command later.

7. Configuration ROM for the Printer subunit

7.1 Conformance

The configuration ROM for the printer subunit shall be compliant with the reference [R7].

The configuration ROM for nodes implementing the printer subunit are expected to implement instance directory and keywords in the configuration ROM, as described in the following clause.

Devices which do not implement instance directories and keywords will not be discovered by controllers searching for devices by instance directories and keywords.

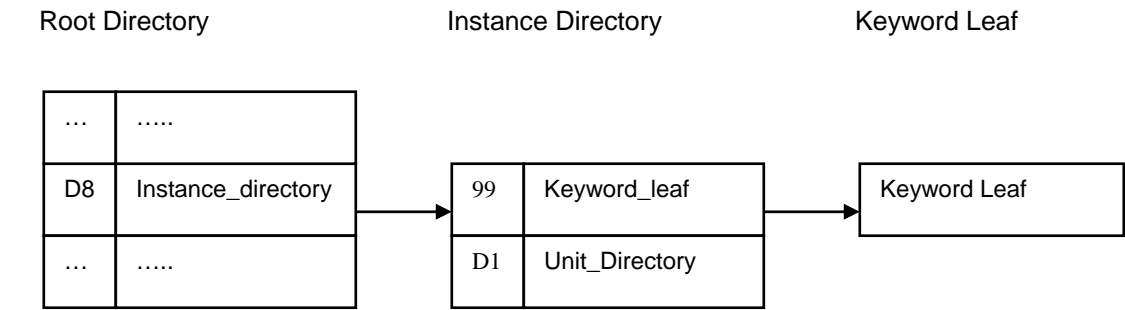
7.2 Instance directories and Keywords

Since printing applications are likely to exist in an multiple protocol environment with a number of printers implementing more than one protocol, some printer controllers will implement/require a protocol independent (not relying on AV/C or other unit software defined device discovery methods) printer discovery and enumeration method.

In order to allow this in an unified manner for all printer devices, an instance directory including a keyword leaf containing the keyword PRINTER is expected to be present in the configuration ROM, per physical instance of the printer implementing the printer subunit. This mechanism will also allow accurate description of the physical instances within a node.

The overall structure of instance directories with keyword leaves is shown in Figure 7-1.

NOTE - More detailed information about the instance directory is defined in Section 7.7.15 of IEEE1212-2000



* other entries are not shown in this figure

Figure 7-1 – Overall Structure of Instance Directories with Keyword leaves

7.2.1 Root directory

The Instance_directory entry with key value D8₁₆ shall be present as an offset to the instance directory.

7.2.2 Instance directory

The following entries are required in the instance directory representing an instance of a printer controlled by the printer subunit within a node ;

- Keyword_Leaf (leaf offset)
- Unit_Directory (directory offset)

The following Figure 7-2 shows an example of an instance directory

Instance_directory_length		CRC	
99	Keyword_leaf (offset)		
D1	Unit_Directory (offset)		
...		

* Multiple Unit_Directory entries may be present in case of an instance supported by multiple protocols.

Figure 7-2 – Instance directory

7.2.2.1 Keyword_leaf entry

The Keyword_leaf entry with key value 99_{16} shall be present as an offset to the keyword leaf which includes the keyword describing the instance controlled by the printer subunit.

7.2.2.2 Unit_Directory entry

The Unit_directory entry with key value $D1_{16}$ shall be present as an offset to the AV/C unit directory

7.2.3 Keyword leaf

The keyword leaf for the instance directory representing an instance controlled by the printer subunit shall include the keyword PRINTER. Other keywords may be included in the keyword leaf.

The following Figure 7-3 shows the format of the keyword leaf containing the keyword PRINTER.

Keyword_leaf_length		CRC	
50_{16} (P)	52_{16} (R)	49_{16} (I)	$4E_{16}$ (N)
54_{16} (T)	45_{16} (E)	52_{16} (R)	00_{16}

Figure 7-3 – Keyword leaf

Annexes

Annex A: Bibliography (informative)

A.1 Bibliography

The following documents provide useful informative information for the reader.

[B1] ANSI X3.159-1989, Programming Language—C.

Annex B: Data transport and connection management

B.1 Connection managements

An AV/C printer is intended to support four types of data transport to capture an print content as follows. For detail, refer to the references listed in the table below.

Table B.1 – Data transports for an AV/C printer subunit

Data transport	Support level	References	AV/C commands required
Asynchronous connections	Mandatory	[R3] and [R4]	AC MANAGE
Isochronous connections	Optional	[R2], [R3] and [R6]	INPUT SELECT
External	Optional	[R3] and [R6]	SIGNAL SOURCE
Other subunits inside an AV unit	Optional	[R3] and [R6]	SIGNAL SOURCE

B.2 Connection setup procedure

The controller shall setup the connection after the print job had been created, prior to issuing the CAPTURE CONTROL command with the subfunction of receive to the printer subunit.

The controller shall confirm the connection is broken before closing the print job in case of asynchronous connection.

Annex C: Command sequence examples (Informative)

C.1 Basic sequence

This section describes the basic sequence to print a single page with default operation mode. As this sequence consists of mandatory commands of the printer subunit specification, all printers compliant with this specification are guaranteed to operate by the sequence described in the table below.

Note that this specification defines that an asynchronous connections shall be supported as mandatory for an AV/C printer subunit, an AV unit which includes an AV/C printer subunit shall support at least the capability of asynchronous connection consumer functionality, as defined in reference [R4] and [R5] .

Table C.1 – Basic sequence for one page print by asynchronous connection

<pre> (1) Find the printer subunit and its version and profiles VERSION STATUS(); (2) Create the print job entry JOB QUEUE CONTROL (add_job, print_job_ID); (3) Establish an asynchronous connection AC MANAGE CONTROL (ALLOCATE); AC MANAGE CONTROL (ATTATCH); (4) Input the images (one images) CAPTURE CONTROL (receive, print_job_ID); (5) Break an asynchronous connection AC MANAGE CONTROL (DETATCH); AC MANAGE CONTROL (RELEASE); (6) Finish the print job JOB QUEUE CONTROL (close_job, print_job_ID); (7) Print finish ~ job disappears </pre>
--

C.2 Optional sequences

C.2.1 Optional sequence for index print with modified operation mode

This section describes the enhanced sequence to control a printer subunit with optional features. Note that the controller is recommended to query the capabilities of a printer subunit before printing, because these features are optional for the printer subunit.

The following table describes an example sequence of printing four images on a page (index print), and generate five copies per page.

Table C.2 – Enhanced sequence for index print

<pre> (1) Find the printer subunit and its version and profiles VERSION STATUS(); (2) Create the print job entry JOB QUEUE CONTROL (add_job, print_job_ID); (3) Set the operation mode of the subunit (index print of four picture, 5 copies) OPERATION MODE(print_job_ID, number_of_pics=4, number_of_copies=5); (4) Establish an asynchronous connection AC MANAGE CONTROL (ALLOCATE); AC MANAGE CONTROL (ATTATCH); (5) Input the images (four images) CAPTURE CONTROL (receive, print_job_ID); CAPTURE CONTROL (receive, print_job_ID); CAPTURE CONTROL (receive, print_job_ID); CAPTURE CONTROL (receive, print_job_ID); (6) Break an asynchronous connection AC MANAGE CONTROL (DETATCH); AC MANAGE CONTROL (RELEASE); (7) Finish the print job JOB_QUEUE CONTROL (close_job, print_job_ID); (8) Polling of the job status (optional) JOB QUEUE STATUS (print_job_ID); (9) Print finish ~ job disappears </pre>

C.2.2 Isochronous capture

This section describes the optional sequence to control a printer subunit using an optional isochronous connection data transport. The isochronous capable printer shall support the INPUT SELECT command defined in [R6]. The sequence is described in the table below.

Table C.3 – Optional sequence for one page print using isochronous connection

<pre> (1) Find the printer subunit and its version and profiles VERSION STATUS(); (2) Create the print job entry JOB QUEUE CONTROL (add_job, print_job_ID); (3) Establish an isochronous connection INPUT SELECT CONTROL (CONNECT); (4) Input the images (one images) CAPTURE CONTROL (receive, print_job_ID); (5) Break an isochronous connection (optional) INPUT SELECT CONTROL (DISCONNECT); (6) Finish the print job JOB_QUEUE CONTROL (close_job, print_job_ID); (7) Polling of the job status (optional) JOB QUEUE STATUS (print_job_ID); (8) Print finish ~ job disappears </pre>

Annex D: Image Data Format Details

D.1 Overview

This section describes details of the following image data formats that AV/C Printer Subunit can handle. Categorization of these formats is shown in Figure D.7-1.

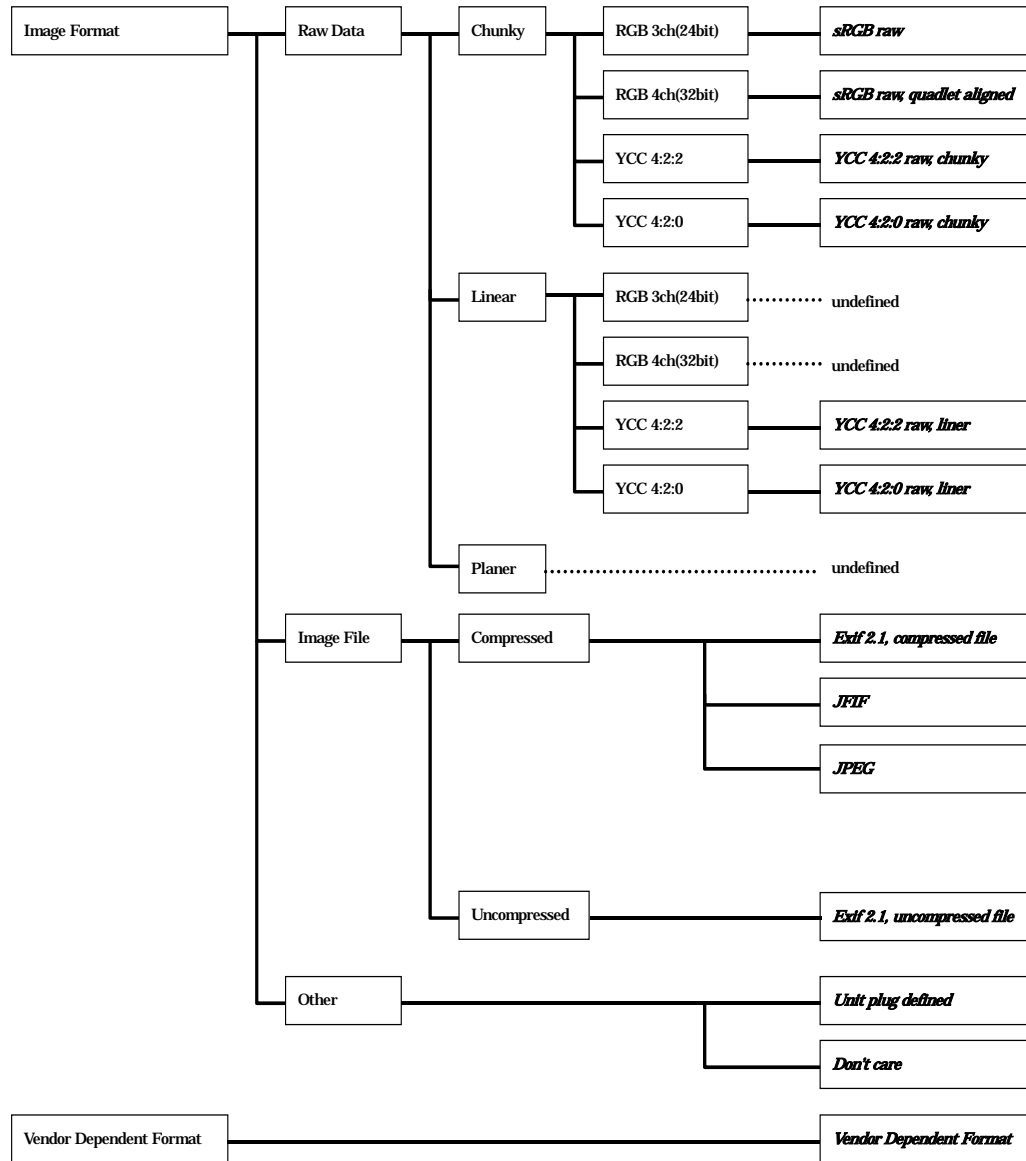


Figure D.7-1 – Categorization of Image Data Format

Categorization of YCC 4:2:0/YCC 4:2:2, chunky/linear formats are shown in Figure D.7-2.

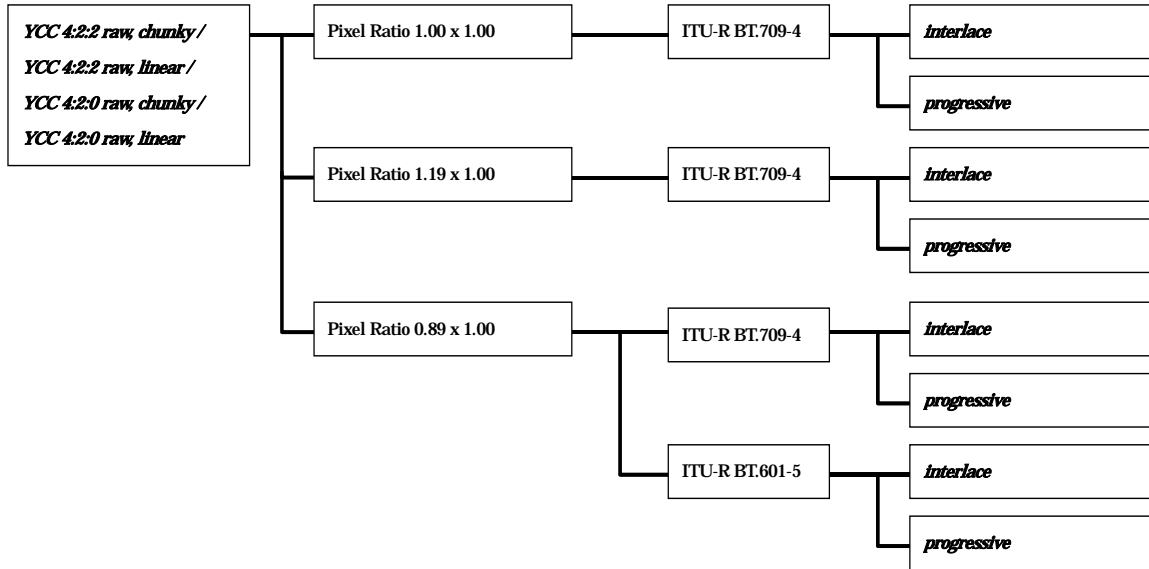


Figure D.7-2 – Categorization of YCC 4:2:0/YCC 4:2:2, chunky/linear formats

D.2 sRGB raw

D.2.1 Image Data Arrangement

Figure D.7-3 shows the image data arrangement of the sRGB raw image data format representing RGB chunky data. One pixel’s data is composed of R(Red), G(Green) and B(Blue) channels and is arranged in order from the starting pixel to the final pixel of raster scanning.

R ₁	G ₁	B ₁	R ₂
G ₂	B ₂	R ₃	G ₃
B ₃	R ₄	G ₄	B ₄
⋮			
R _N	G _N	B _N	

Where: Ri: i th Data for red channel (i=1..N)
 Gi: i th Data for green channel (i=1..N)
 Bi: i th Data for blue channel (i=1..N)
 N: Total number of pixels
 (i=1 represents the starting pixel and i=N represents the final pixel of raster scanning)

Figure D.7-3 – Image Data Arrangement of sRGB raw

D.2.2 Bit Depth of Image Data

All channels are composed of 8 bit depth image data.

D.2.3 Color Space

The RGB color space shall confirm to [R8], for details.

D.3 sRGB raw, quadlet aligned

D.3.1 Image Data Arrangement

Figure D.7-4 shows the image data arrangement of sRGB raw, quadlet aligned image data format representing RGB chunky data with dummy data for quadlet allocation. One pixel's data is composed of D(Dummy), R(Red), G(Green) and B(Blue) channels and is arranged in order from the starting pixel to the final pixel of raster scanning.

00 ₁₆ (Dummy)	R ₁	G ₁	B ₁
00 ₁₆ (Dummy)	R ₂	G ₂	B ₂
00 ₁₆ (Dummy)	R ₃	G ₃	B ₃
⋮			
00 ₁₆ (Dummy)	R _N	G _N	B _N

Where:

- R_i: ith Data for red channel (i=1..N)
- G_i: ith Data for green channel (i=1..N)
- B_i: ith Data for blue channel (i=1..N)
- N: Total number of pixels
(i=1 represents the starting pixel and i=N represents the final pixel of raster scanning)

Figure D.7-4 – Image Data Arrangement of sRGB raw, quadlet aligned

D.3.2 Bit Depth of Image Data

All channels including the dummy channel are composed of 8 bit depth image data.

D.3.3 Color Space

The RGB color space shall conform to [R8], for details.

D.4 YCC 4:2:2 raw, chunky

D.4.1 Image Data Arrangement

Figure D.7-5 shows the image data arrangement of the YCC 4:2:2 raw, chunky image data format. One pixel's data is composed of Y, Cb and Cr channels and is arranged in order from the starting pixel to the final pixel of raster scanning. Samples of Cb, Cr are half of Y Channel.

$Y_1(L_1)$	$Y_2(L_1)$	$Cb_1(L_1)$	$Cr_1(L_1)$
$Y_3(L_1)$	$Y_4(L_1)$	$Cb_3(L_1)$	$Cr_3(L_1)$
⋮			
$Y_{N-1}(L_1)$	$Y_N(L_1)$	$Cb_{N-1}(L_1)$	$Cr_{N-1}(L_1)$
$Y_1(L_2)$	$Y_2(L_2)$	$Cb_1(L_2)$	$Cr_1(L_2)$
⋮			
$Y_{N-1}(L_M)$	$Y_N(L_M)$	$Cb_{N-1}(L_M)$	$Cr_{N-1}(L_M)$

Where: $Y_i(L_j)$: Y Data of i th pixel in the j th line ($i=1..N, j=1..M$)
 $Cb_i(L_j)$: Cb Data of i th pixel in the j th line ($i=1, 3, 5..N-1, j=1..M$)
 $Cr_i(L_j)$: Cr Data of i th pixel in the j th line ($i=1, 3, 5..N-1, j=1..M$)
 N: Total number of pixels in the line (it should be even)
 M: Total number of lines in the plane

Figure D.7-5 – Image Data Arrangement of YCC 4:2:2 raw, chunky

D.4.2 Bit Depth of Image Data

All channels are composed of 8 bit depth image data.

D.4.3 Color Parameters

There are two types of color parameters. The YCC (ITU-R BT.601-5) color parameters shall confirm to [R9] for details. The YCC (ITU-R BT.709-4) color parameters shall confirm to [R10] for details.

D.5 YCC 4:2:0 raw, chunky

D.5.1 Image Data Arrangement

Figure D.7-6 shows the image data arrangement of the YCC 4:2:0 raw, chunky image data format. One pixel’s data is composed of Y, Cb and Cr channels and is arranged in order from the starting pixel to the final pixel of raster scanning. Samples of Cb, Cr are quarter of Y Channel.

$Y_1(L_1)$	$Y_2(L_1)$	$Y_1(L_2)$	$Y_2(L_2)$
$Cb_1(L_1)$	$Cr_1(L_1)$	$Y_3(L_1)$	$Y_4(L_1)$
$Y_3(L_2)$	$Y_4(L_2)$	$Cb_3(L_1)$	$Cr_3(L_1)$
⋮			
$Y_{N-3}(L_{M-1})$	$Y_{N-2}(L_{M-1})$	$Y_{N-3}(L_M)$	$Y_{N-2}(L_M)$
$Cb_{N-3}(L_{M-1})$	$Cr_{N-3}(L_{M-1})$	$Y_{N-1}(L_{M-1})$	$Y_N(L_{M-1})$
$Y_{N-1}(L_M)$	$Y_N(L_M)$	$Cb_{N-1}(L_{M-1})$	$Cr_{N-1}(L_{M-1})$

Where: $Y_i(L_j)$: Y Data of i th pixel in the j th line ($i=1..N, j=1..M$)
 $Cb_i(L_j)$: Cb Data of i th pixel in the j th line ($i=1, 3, 5..N-1, j=1,3,5 ..M-1$)
 $Cr_i(L_j)$: Cr Data of i th pixel in the j th line ($i=1, 3, 5..N-1, j=1,3, 5..M-1$)
 N: Total number of pixels in the line (it should be even)
 M: Total number of lines in the plane (it should be even)

Figure D.7-6 – Image Data Arrangement of YCC 4:2:0 raw, chunky

D.5.2 Bit Depth of Image Data

All channels are composed of 8 bit depth image data.

D.5.3 Color Parameters

There are two types of color parameters. The YCC (ITU-R BT.601-5) color parameters shall confirm to [R9] for details. The YCC (ITU-R BT.709-4) color parameters shall confirm to [R10] for details.

D.6 YCC 4:2:2 raw, linear

D.6.1 Image Data Arrangement

Figure D.7-7 shows the image data arrangement of the YCC 4:2:2 raw, linear image data format. One pixel's data is composed of Y, Cb and Cr channels and is arranged in order from the starting pixel to the final pixel of raster scanning in each channel. Samples of Cb, Cr are half of Y Channel.

$Y_1(L_1)$	$Y_2(L_1)$	$Y_3(L_1)$	$Y_4(L_1)$
⋮			
$Y_{N-3}(L_1)$	$Y_{N-2}(L_1)$	$Y_{N-1}(L_1)$	$Y_N(L_1)$
$Cb_1(L_1)$	$Cr_1(L_1)$	$Cb_3(L_1)$	$Cr_3(L_1)$
⋮			
$Cb_{N-3}(L_1)$	$Cr_{N-3}(L_1)$	$Cb_{N-1}(L_1)$	$Cr_{N-1}(L_1)$
$Y_1(L_2)$	$Y_2(L_2)$	$Y_3(L_2)$	$Y_4(L_2)$
⋮			
$Cb_{N-3}(L_M)$	$Cr_{N-3}(L_M)$	$Cb_{N-1}(L_M)$	$Cr_{N-1}(L_M)$

Where: $Y_i(L_j)$: Y Data of i th pixel in the j th line ($i=1..N, j=1..M$)
 $Cb_i(L_j)$: Cb Data of i th pixel in the j th line ($i=1, 3, 5..N-1, j=1..M$)
 $Cr_i(L_j)$: Cr Data of i th pixel in the j th line ($i=1, 3, 5..N-1, j=1..M$)
 N : Total number of pixels in the line (it should be even)
 M : Total number of lines in the plane

Figure D.7-7 – Image Data Arrangement of YCC 4:2:2 raw, linear

D.6.2 Bit Depth of Image Data

All channels are composed of 8 bit depth image data.

D.6.3 Color Parameters

There are two types of color parameters. The YCC (ITU-R BT.601-5) color parameters shall confirm to [R9] for details. The YCC (ITU-R BT.709-4) color parameters shall confirm to [R10] for details.

D.7 YCC 4:2:0 raw, linear

D.7.1 Image Data Arrangement

Figure D.7-8 shows the image data arrangement of the YCC 4:2:0 raw, linear image data format. One pixel's data is composed of Y, Cb and Cr channels and is arranged in order from the starting pixel to the final pixel of raster scanning in each channel. Samples of Cb, Cr are quarter of Y Channel.

$Y_1(L_1)$	$Y_2(L_1)$	$Y_3(L_1)$	$Y_4(L_1)$
⋮			
$Y_{N-3}(L_1)$	$Y_{N-2}(L_1)$	$Y_{N-1}(L_1)$	$Y_N(L_1)$
$Y_1(L_2)$	$Y_2(L_2)$	$Y_3(L_2)$	$Y_4(L_2)$
⋮			
$Y_{N-3}(L_2)$	$Y_{N-2}(L_2)$	$Y_{N-1}(L_2)$	$Y_N(L_2)$
$Cb_1(L_1)$	$Cr_1(L_1)$	$Cb_3(L_1)$	$Cr_3(L_1)$
⋮			
$Cb_{N-3}(L_1)$	$Cr_{N-3}(L_1)$	$Cb_{N-1}(L_1)$	$Cr_{N-1}(L_1)$
$Y_1(L_3)$	$Y_2(L_3)$	$Y_3(L_3)$	$Y_4(L_3)$
⋮			
$Cb_{N-3}(L_{M-1})$	$Cr_{N-3}(L_{M-1})$	$Cb_{N-1}(L_{M-1})$	$Cr_{N-1}(L_{M-1})$

Where: $Y_i(L_j)$: Y Data of i th pixel in the j th line ($i=1..N, j=1..M$)
 $Cb_i(L_j)$: Cb Data of i th pixel in the j th line ($i=1, 3, 5..N-1, j=1, 3, 5..M-1$)
 $Cr_i(L_j)$: Cr Data of i th pixel in the j th line ($i=1, 3, 5..N-1, j=1, 3, 5..M-1$)
 N : Total number of pixels in the line (it should be even)
 M : Total number of lines in the plane (it should be even)

Figure D.7-8 – Image Data Arrangement of YCC 4:2:0 raw, linear

D.7.2 Bit Depth of Image Data

All channels are composed of 8 bit depth image data.

D.7.3 Color Parameters

There are two types of color parameters. The YCC (ITU-R BT.601-5) color parameters shall conform to [R9] for details. The YCC (ITU-R BT.709-4) color parameters shall conform to [R10] for details.

D.8 Exif 2.1 compressed file

D.8.1 Image File Format

Image file format shall conform to Exif Ver.2.1 See [R11] for details.

D.8.2 Additional Specification

Table D.4 shows additional restrictions to the choices defined in Exif Ver.2.1 specification.

Target device supporting Exif as image data format shall be capable of handling Exif image files with any variations shown in this figure.

Table D.4 – Additional Restrictions of Exif 2.1

Data Format	JPEG Compression
Pixel Sampling	YCC 4:2:0 or 4:2:2
Restart Marker	May be inserted
Image Width, Image Length	Shall be the same value as image_size_x and image_size_y in the CAPTURE CONTROL command frame
Color Space	sRGB
Byte Order	Little Endian or Big Endian

D.8.2.1 Exif Tag Handling exceptions

Several tags defined in the Exif specification need to be handled as shown in Table D.5.

Table D.5 – Exif Tag handling

Tag	Handling
Orientation Tag	The target may ignore the Orientation Tag included in the Exif image file to generate the print out.
Xresolution Tag Yresolution Tag ResolutionUnit Tag	The target may ignore the Xresolution, Yresolution and ResolutionUnit tags included in the Exif image file to generate the print out.

D.8.3 Image Data Arrangement

Table D.9 shows the image data arrangement of transmitting an Exif image file.

D ₁	D ₂	D ₃	D ₄
D ₅	D ₆	D ₇	D ₈
⋮			
D _{N-1}	D _N		

Where: Di: i th data element of Exif image file (i=1..N)
 N: Total data size of Exif image file(bytes)

Figure D.7-9 – Image Data Arrangement of Exif

D.9 JFIF

D.9.1 Image File Format

Image File Format shall conform to JFIF v1.02. See [R12] for details.

D.9.2 Additional Specification

Table D.6 shows additional restrictions to the choices defined in JFIF v1.02 Specification.

Target supporting JFIF as image data format shall be capable of handling JFIF image files with any variations shown in this figure.

Table D.6 – Additional Restrictions of JFIF

Data Format	JPEG Baseline ADCT
Pixel Sampling	YCC 4:2:0 or 4:2:2
DHT marker	up to 4
DQT marker	up to 3
DQT tables	1 table per DQT or single DQT defines all tables
DHT tables	1 table per DHT or single DHT defines all tables
X,Y density in APP0	The target may reflect X and Y density in printing if units for X and Y density is defined.
Color Space	recommended use of sRGB in printing

D.9.3 Image Data Arrangement

Figure D.7-10 shows the image data arrangement of transmitting JFIF image file.

D ₁	D ₂	D ₃	D ₄
D ₅	D ₆	D ₇	D ₈
⋮			
D _{N-1}	D _N		

Where: D_i: i th data element of JFIF image file (i=1..N)
 N: Total data size of JFIF image file(bytes)

Figure D.7-10 – Image Data Arrangement of JFIF

D.10 Exif 2.1, Uncompressed file

D.10.1 Image File Format

Image file format shall conform to Exif Ver.2.1. as listed below.

- 1) RGB uncompressed data: Baseline TIFF Rev. 6.0 RGB Full Color Images
- 2) YCbCr uncompressed data: TIFF Rev. 6.0 Extensions YCbCr Images,
 See “2.5.2” and “2.5.3” of [R11] for detail.

D.10.1.1 Additional Specification

The following Table D.7 shows additional restrictions to the choices defined in Exif Ver.2.1 specification, uncompressed file.

Table D.7 – Additional Restrictions of Exif 2.1

Pixel Sampling in case of YCbCr	YCC 4:2:0 or 4:2:2
Color Space	SRGB
Byte Order	Little Endian or Big Endian

D.10.1.2 Exif Tag Handling exceptions

Several tags defined in the Exif specification need to be handled as shown in Table D.8 below.

Table D.8 – Exif Tag handling

Tag	Handling
Orientation Tag	The target may ignore the Orientation Tag included in the Exif image file to generate the print out.
Xresolution Tag Yresolution Tag ResolutionUnit Tag	The target may ignore the Xresolution, Yresolution and ResolutionUnit tags included in the Exif image file to generate the print out.

D.10.2 Image Data Arrangement

The following Figure D.7-11 shows the image data arrangement of transmitting an Exif image file.

D ₁	D ₂	D ₃	D ₄
D ₅	D ₆	D ₇	D ₈
⋮			
D _{N-1}	D _N		

Where: D_i: i th data element of Exif image file (i=1..N)
 N: Total data size of Exif image file(bytes)

Figure D.7-11 – Image Data Arrangement of Exif

D.11 JPEG

D.11.1 Image File Format

Image file format shall conform to JPEG baseline ADCT. See [R13] for detail.

D.11.2 Additional Specification

Table D.9 shows additional restrictions to the choices defined in JPEG specification.

Table D.9 – Additional Restrictions of JPEG

Data Format	JPEG Baseline ADCT
Pixel Sampling	YCC 4:2:0 or 4:2:2
DHT marker	up to 4
DQT marker	up to 3
DQT tables	1 table per DQT or single DQT defines all tables
DHT tables	1 table per DHT or single DHT defines all tables
Color Space	recommended use of sRGB in printing

D.11.3 Image Data Arrangement

Figure D.7-12 shows the image data arrangement of transmitting JPEG image file.

D ₁	D ₂	D ₃	D ₄
D ₅	D ₆	D ₇	D ₈
⋮			
D _{N-1}	D _N		

Where: Di: i th data element of JPEG image file (i=1..N)
 N: Total data size of JPEG image file(bytes)

Figure D.7-12 – Image Data Arrangement of JPEG

Annex E: Configuration ROM example (informative)

E.1 ROM example

		unit	:hexadecimal				
		Rom Base Address	:FFFF F000 0400 ₁₆				
		value of data				address	
		1Quadlet = 4Byte / entry field					
bus info block	bus_info_length	crc_length	bus_info_crc				04 04 EA BF 0400
	"1"	"3"	"9"		"4"		31 33 39 34 0404
	lrmc crmc lsc bmc pmc reserved	cyc_clk_acc	max_rec	r	max_ROM	generation r link_spd	E0 64 61 02 0408
	node_vendor_ID			chip_ID_hi			FF FF FF FF 040C
root directory	chip_ID_lo					FF FF FF FF 0410	
	directory_length	crc				00 07 B4 54 0414	
	03	vendor_ID				03 FF FF FF 0418	
	81	textual descriptor leaf offset				81 00 00 0E 041C	
	17	model_ID				17 FF FF FF 0420	
	81	textual descriptor leaf offset				81 00 00 12 0424	
	0C	node_capabilities				0C 00 83 C0 0428	
	D1	unit directory offset				D1 00 00 05 042C	
instance directory	D8	instance directory offset				D8 00 00 01 0430	
	directory_length	crc				00 02 44 2D 0434	
	99	keyword leaf offset				99 00 00 13 0438	
unit directory	D1	unit directory offset				D1 00 00 01 043C	
	directory_length	crc				00 04 BD 22 0440	
	12	specifier_ID				12 00 A0 2D 0444	
	13	version				13 01 00 01 0448	
	17	model_ID				17 FF FF FF 044C	
descriptor leaf for vendor name	81	textual descriptor leaf offset				81 00 00 07 0450	
	leaf_length	crc				00 05 C9 15 0454	
	descriptor_type	specifier_ID				00 00 00 00 0458	
	width	character_set	language			00 00 00 00 045C	
	"V"	"e"	"n"	"d"		56 65 6E 64 0460	
	"o"	"r"	" "	"N"		6F 72 20 4E 0464	
descriptor leaf for model name	"a"	"m"	"e"	Null		61 6D 65 00 0468	
	leaf_length	crc				00 05 7F 16 046C	
	descriptor_type	specifier_ID				00 00 00 00 0470	
	width	character_set	language			00 00 00 00 0474	
	"M"	"o"	"d"	"e"		4D 6F 64 65 0478	
	"l"	" "	"N"	"a"		6C 20 4E 61 047C	
	"m"	"e"	Null	Null		6D 65 00 00 0480	
keyword leaf	leaf_length	crc				00 02 9C 68 0484	
	"P"	"R"	"l"	"N"		50 52 49 4E 0488	
	"T"	"E"	"R"	Null		54 45 52 00 048C	

The figure below shows the bus information block, root directory, unit directory and instance directory for an AV/C device compliant with the printer subunit specification.

Annex F: Result code index (informative)

The following table is the master index of the result values defined in this document.

Value	Symbol	Meaning	Related commands
00 ₁₆	no_error	The command execution had been finished successfully, or no command is under execution.	JOB QUEUE OPERATION MODE CAPTURE CAPTURE REF PRINTER STATUS
80 ₁₆	format_error	The command execution had been aborted, because the print content format is not supported or unacceptable now.	CAPTURE CAPTURE REF
81 ₁₆	image_size_error	The command execution had been aborted, because the image size is not supported or unacceptable now.	CAPTURE
82 ₁₆	data_size_error	The command execution had been aborted, because the data length is not supported or unacceptable now.	CAPTURE
83 ₁₆	invalid_format	The command execution had been aborted, because the print content format is not the same as specified by the command frame.	CAPTURE CAPTURE REF
84 ₁₆	invalid_image_size	The command execution had been aborted, because the image size is not the same as specified by the command frame.	CAPTURE
85 ₁₆	invalid_data_size	The command execution had been aborted, because the receiving data had been lost or the receiving data is bigger than specified.	CAPTURE
86 ₁₆	no_job_exists	The print job with specified print_job_ID or job_position does not exist.	JOB QUEUE OPERATION MODE CAPTURE CAPTURE REF PRINTER STATUS
87 ₁₆	busy	The command is rejected because the subunit is busy.	JOB QUEUE OPERATION MODE CAPTURE CAPTURE REF PRINTER STATUS
88 ₁₆	no_process_to_abort	The command is rejected because currently there is no capturing process undergoing.	CAPTURE CAPTURE REF
89 ₁₆	illegal_sequence	The command is rejected because the print content capturing already started.	OPERATION MODE
8A ₁₆	unsupported_parameter	The values of the requested operation_mode_parameters are not supported by the subunit.	OPERATION MODE
90 ₁₆	no_pending_process	The resume subfunction is rejected, because there is no pending capturing process. The capturing may be finished.	CAPTURE CAPTURE REF
91 ₁₆	aborted	The command is aborted because the controller issued the CAPTURE CONTROL command with the subfunction of abort.	CAPTURE CAPTURE REF
92 ₁₆	queue_full	The command is rejected because the job queue is full.	JOB QUEUE OPERATION MODE CAPTURE CAPTURE REF PRINTER STATUS
93 ₁₆	job_already_exists	The add_job subfunction is rejected because the specified job already exist.	JOB QUEUE
FE ₁₆	any_other_error	The command execution had been aborted, because any other error occurred.	JOB QUEUE OPERATION MODE CAPTURE CAPTURE REF PRINTER STATUS
FF ₁₆	-	No meaning	JOB QUEUE OPERATION MODE CAPTURE CAPTURE REF PRINTER STATUS
Other values	-	Reserved	JOB QUEUE OPERATION MODE CAPTURE CAPTURE REF PRINTER STATUS

Annex G: Print By Reference logical System model for markup data pull print profile (informative)

The following figure indicates the Print By reference logical model for “markup data pull print profile”

In this case Data source unit has AV/C camera storage subunit and . camera storage subunit acts as data source subunit.

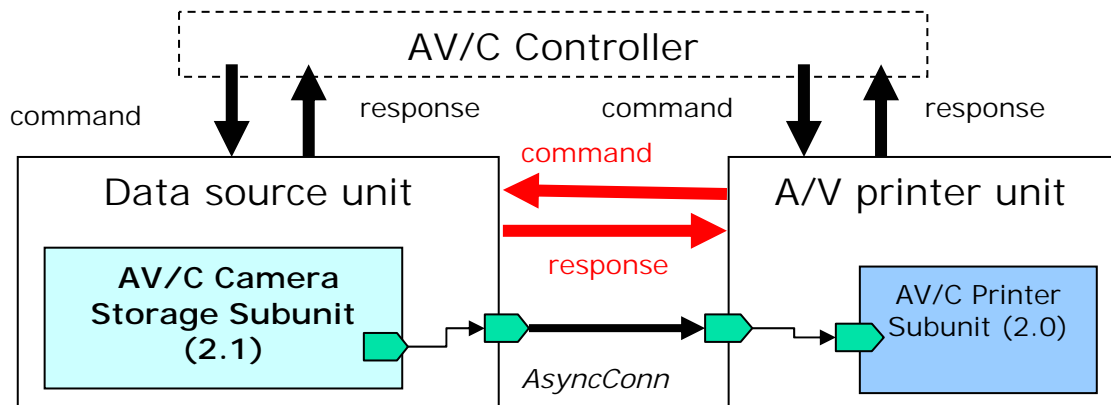


Figure G.1 – Print By Reference logical System Model for markup pull print profile

On the Print By reference logical model for “markup data pull print profile”, AV/C Camera Storage Subunit 2.1 acts as a data source subunit in data source unit.

For this case , the data retrieving sequence after establishing asynchronous connection is ;

1. The controller issues CAPTURE REF CONTROL command to the printer subunit 2.0 in A/V printer unit. The profile of the printer subunit shall be ‘markup data pull profile.’ at least .
2. The AV/Printer unit acts as controller for camera storage subunit. The controller uses the *subunit_type* field value for specifying type of data source unit. The controller uses *subunit_ID* and *producer_node_ID* fields for specifying camera storage. subunit. The controller uses *destination_plug* field for specifying asynchronous connection to be used for data transfer.
3. The AV/Printer unit issues SEND FILE CONTROL command or SEND FILE PARTIALcommand for retrieving the specified markup language data . SEND FILE CONTROL command or SEND FILE PARTIALcommand is issued with *source_plug* and *absolute_file_path* retrieved from CAPTURE REF CONTROL command. In this case , the *psysical_volume_number* and *logical_volume_number* shall be the first volume(00₁₆).
4. After retrieving the specified markup language data , the AV/Printer unit parses it .If it refers the image object data inside, AV/Printer unit resolves *absolute_file_path* of the image object data. Then, AV/Printer unit retrieves the image object data in the same way for retrieving markup language data

Annex H: Command sequence examples for markup data pull print profile (informative)

H.1 Basic sequence

This section describes the basic sequence to print a markup language data with default operation mode. As this sequence consists of mandatory commands of the printer subunit specification for markup data pull print profile, all printers compliant with this specification are guaranteed to operate by the sequence described in the table below.

Note that this specification defines that an asynchronous connections shall be supported as mandatory for an AV/C printer subunit, an AV unit which includes an AV/C printer subunit shall support at least the capability of asynchronous connection consumer functionality, as defined in reference [R4] and [R5] .

Regardless of whether markup language data refers image data(s) or not , this basic sequence is same way.

Table H.1 – Basic sequence for markup data pull print profile by asynchronous connection

<pre> (1)Find the printer subunit and its version and profiles VERSION STATUS(); (2)Create the print job entry JOB QUEUE CONTROL (add_job, print_job_ID); (3)Establish an asynchronous connection AC MANAGE CONTROL (ALLOCATE); AC MANAGE CONTROL (ATTATCH); (4)Input the markup language data CAPTURE CONTROL REF (retrieve, print_job_ID); (5)Break an asynchronous connection AC MANAGE CONTROL (DETATCH); AC MANAGE CONTROL (RELEASE); (6)Finish the print job JOB QUEUE CONTROL (close_job, print_job_ID); (7)Print finish ~ job disappears </pre>
--

H.2 Constraint

For “markup data pull print profile”, only one markup language data is inputted per one print job.