PictBridge

CIPA DC-001-*2003 Rev. 2.0*

Digital Photo Solutions for Imaging Devices

Disclaimer and Copyright Notice

THIS DRAFT DOCUMENT IS PROVIDED “AS IS” WITH NO WARRANTIES WHATSOEVER, INCLUDING ANY WARRANTY OF MERCHANTABILITY, NONINFRINGEMENT, FITNESS FOR ANY PARTICULAR PURPOSE, OR ANY WARRANTY OTHERWISE ARISING OUT OF ANY PROPOSAL, SPECIFICATION OR SAMPLE. All liability, including liability for infringement of any proprietary rights, relating to use of information in this document is disclaimed. No license, express or implied, by estoppel or otherwise, to any intellectual property rights are granted herein.

This document is a draft for comment only and is subject to change without notice. Readers shall not design products based on this document.

Revision History

Rev	Date	Comments	Page
1.0	February 3, 2003	Published Revision 1.0.	
2.0	November 27, 2006	Published Revision 2.0.	
		● Added the scope for DPS Version 1.1.	1
		● Added 3.5 Symbol for parameters incompatible with DPS1.0.	3
		● Added how to enumerate the list of DPS versions.	15,16
		● Added the description concerning the `charRepertoires` parameter.	17,49,50,53,61,65,66
		● Incorporated the Errata for "CIPA DC-001-2003 February 3".	21,22,32,57,59,60,70,104,105
		● Added the description for XHTML-Print.	21,22,26,27,32,37,38,108,110
		● Added the description for a rule to add new parameter values.	53
		● Added 10.5.2 fileTypes Minor Code.	64,65
		● Updated Section 12 References.	80,81
		● Added Appendix E. E.2 Print Result Compatibility.	111
		● Added Appendix G. pictbridge URI scheme.	113,114
		● Added Appendix H. DPS Support for XHTML-Print.	115-121

Abstract

This standard defines a set of protocols and operations that enable an image source device, such as a Digital Still Camera, to create and send a print job to an image output device, such as a Printer. Support is provided for printing a single image or a collection of images and for providing the user with ongoing status results.

This standard has been designed to support popular image formats used in Digital Still Cameras, including the Exif formats and the Digital Print Order Format (DPOF). See [Exif] and [DPOF] for more information. This specification includes conformance requirements for both cameras and printers, providing a strong basis for wide interoperability.

Contributors/Reviewers for CIPA DC-001-2003

Contributors

Aichi, Takao	CANON INC.
Anderson, Dean	Hewlett-Packard Company
Bouno, Hironori	Hewlett-Packard Company
Endo, Masakatsu	SEIKO EPSON CORPORATION
Goto, Fumihiro	CANON INC.
Grant, Melinda	Hewlett-Packard Company
Hirabayashi, Hiromitsu Hank	CANON INC.
Katayama, Toshihiko	SEIKO EPSON CORPORATION
Kawamura, Hideaki	CANON INC.
Kinoshita, Takao	CANON INC.
Mikami, Ruriko	CANON INC.
Morozumi, Hideki	SEIKO EPSON CORPORATION
Nishizawa, Katsuhiko	SEIKO EPSON CORPORATION
Ogiwara, Satoshi	CANON INC.
Okumoto, Koji	Sony Corporation
Sakamoto, Kazuya	CANON INC.
Sakuda, Kenji	SEIKO EPSON CORPORATION
Sakurada, Nobuaki	CANON INC.
Shults, Gerrie	Hewlett-Packard Company
Suenaga, Kazunori	SEIKO EPSON CORPORATION
Tanimoto, Wataru	Sony Corporation
Wakai, Yoichi	SEIKO EPSON CORPORATION
Williams, Eric	Hewlett-Packard Company
Yamada, Akitoshi	CANON INC.
Yamaya, Kunihiko	Sony Corporation
Yamazaki, Junichi	Hewlett-Packard Company
Yano, Kentaro	CANON INC.

Reviewers

Kokubun, Kouzou	FUJI PHOTO FILM CO.,LTD.
Kuba, Hirokazu	OLYMPUS OPTICAL CO.,LTD.
Nakamura, Toshiya	OLYMPUS OPTICAL CO.,LTD.
Suzuki, Takashi	OLYMPUS OPTICAL CO.,LTD.
Tanaka, Hiroshi	FUJI PHOTO FILM CO.,LTD.
Ueda, Satoshi	FUJI PHOTO FILM CO.,LTD.
Watanabe, Mikio	FUJI PHOTO FILM CO.,LTD.
Yasuda, Tomonaga	OLYMPUS OPTICAL CO.,LTD.
Yoda, Yoshiro	OLYMPUS OPTICAL CO.,LTD.

Contributors/Reviewers for CIPA DC-001-2003 Rev. 2.0

Abe, Koji	CANON INC.
Aichi, Takao	CANON INC.
Fukushima, Keito	Olympus Corporation
Garrido, Ramon	Hewlett-Packard Company
Ichimura, Kenji	Olympus Corporation
Kawamura, Hideaki	CANON INC.
Kitahara, Yoshinao	SEIKO EPSON CORPORATION
Morozumi, Hideki	SEIKO EPSON CORPORATION
Nakamura, Satoshi	FUJIFILM Corporation
Nishizawa, Katsuhiko	SEIKO EPSON CORPORATION
Ogiwara, Satoshi	CANON INC.
Okumoto, Koji	Sony Corporation
Sakuda, Kenji	SEIKO EPSON CORPORATION
Sakurada, Nobuaki	CANON INC.
Shults, Gerrie	Hewlett-Packard Company
Watanabe, Mikio	FUJIFILM Corporation
Yamaya, Kunihiko	Sony Corporation
Yamazaki, Junichi	Hewlett-Packard Company
Yoda, Yoshiro	Olympus Imaging Corp.

Contents

1. Document Status

This section describes the status of this document at the time of its publication. Other documents may supersede this document.

This is a CIPA Standards specification. This specification is intended for product design. Implementations should ensure compliance with this specification. This specification exists in two parts—this document and *DPS parameter values addition supplements for "CIPA DC-001-2003 Rev. 2.0 or later"*. For more information about the supplements, see section 10 and [Supplements].

2. Scope

2.1 Introduction

This specification defines a protocol enabling interoperability between a digital still photography device as typified by a Digital Still Camera (DSC) and an image output device as typified by a Printer. The protocol defines a set of capabilities in terms of operations and their parameters, and specifies an unambiguous wire protocol for their communication.

This specification is motivated by the current lack of standard specifications for direct communication between DSCs and printers. Currently in the marketplace, such interoperability is based on proprietary solutions, and it is not possible for users to print directly from their DSC to another company's printer. This specification aims to remove that limitation, and thereby improve the usability of digital photography.

All defined features are process-mandatory. This means that if a feature is used, it shall be used in the manner specified by this document. Whether a particular feature is mandatory or optional for compliance is stated separately for each side of the interface.

Targeted usage models for Version 1.0 include simple one-button printing of a single image, the printing of several copies of an image on a page, printing a collection of images, and printing images at specified sizes. Exacting layout of different sized images on a single page is not supported in Version 1.0.

In DPS Version 1.1, how to process XHTML-Print (see [XHTML-Print]) is clarified and new parameter values are added in *DPS parameter values addition supplements for "CIPA DC-001-2003 Rev. 2.0 or later"* (see [Supplements]). These are all optional and available between DPS Version 1.1 compliant devices. DPS Version 1.1 is interoperable with DPS Version 1.0. The DPS Version of DPS product that is implemented to meet this document shall be Version 1.1.

When the document refers to an image output device, a color printer typically used in the home for photo printing may be considered as representative if no additional explanation is provided.

2.2 Design Objectives

As this specification is targeted at cost-sensitive, mass-market devices, it must be implementable in a relatively small memory footprint and with limited CPU resources.

The protocol defined by this document defines application layer functionality, and as much as possible, independence from underlying session and transport layers is maintained. This specification defines an over-the-wire protocol that is platform and API independent. Consistent with the broad industry investment in [XML], it is used to describe the interface.

To enable a fast time-to-market, complexity is minimized and existing standards are used or leveraged.

Usability is a key aspect of this targeted solution, as both DSC's and Printers are mass-market devices. Manufacturers undertaking implementation of either side of the interface are strongly encouraged to provide the best possible user feedback on the overall printing process. Specifically, the user should be able to discern:

- When a connection has been made or has been lost between the two devices.
- When the devices are communicating appropriately, i.e., “Things are working”...
- When a problem has occurred; for common problems (e.g., paper out), the user should be able to determine what has happened and how to recover and continue.

The first-generation solution will be deployed over USB, utilizing the Still Image Capture Device class. This class requires the support of the ISO 15740:2005 Picture transfer protocol (PTP); DPS is designed such that much of its functionality may be directly mapped to an existing PTP implementation, thereby reducing the investment in a print application for a DSC. See [USB], [SICD] and [PTP] for more information.

It is desirable to provide the flexibility to enable the user to control the printing operation from a user interface provided by either the DSC or the Printer. For this version, the user will initiate and control the print job from the DSC; however, this specification introduces a model to determine dynamically where this control should reside, with the intent to offer this versatility in future versions.

3. Symbols and Conventions

3.1 Conformance Status Symbols

In this document the following symbols are used:

"**M**" means that support for the feature is mandatory (used for capabilities that *shall* be used in the protocol).

"**R**" means that support for the feature is recommended (used for capabilities that *should* be used in the protocol).

"**O**" means that support for the feature is optional (used for capabilities that *may* be used in the protocol).

3.2 Naming Conventions

Action and parameter names in general are mixed case with the first letter of interior words being capitalized. Operation and event names begin with "DPS_". Parameter names begin with a lower-case character and contain only alphabetic characters.

3.3 Numeric Values

Numeric values are encoded and transmitted as US-ASCII strings representing hexadecimal or decimal values. Hexadecimal strings are not preceded by "0x" or "0X". Unless otherwise stated, all hexadecimal strings shall be exactly 8 characters in length; if the corresponding numeric value could be represented in fewer than 8 characters, the string shall be padded with leading zero characters. Also, unless otherwise stated, decimal value strings shall be exactly 3 digits in length; and, if the corresponding numeric value could be represented in fewer than 3 digits, the string shall be padded with leading zero characters.

For hexadecimal and decimal values that are specified as integers, only digits and hexadecimal characters shall be used; i.e., an integer value shall not be preceded by a plus or minus sign or by a decimal point, and shall not contain commas or decimal points. For example, "005" is a valid integer value, whereas "+005", "+05", "5.00", "05." and "5,000" are invalid DPS integer values.

When converting an ASCII string into a numeric value, network or big endian byte ordering shall be used.

(“**”, “****”, “******”, and “********” mean hidden values.)

3.4 Conformance Language

The keywords “shall”, “should”, “may”, and “can” in this document are to be interpreted as described in the *IEEE Standards Style Manual*, see [Style].

3.5 Symbol for parameters incompatible with DPS1.0

V1.1 means that parameters incompatible with DPS Version 1.0. Those parameters shall not be sent to DPS Version 1.0 compliant devices.

4. Terminology

DPS Acronym of "Digital Photo Solutions for imaging devices".

DPS Version 1.1 The specifications of this standard.

DSC Digital Still Image Camera.

event A state change in one device which is communicated to another device, usually asynchronously.

fileID Unique 32-bit identifier for each DPS object. Usually represented in this document as an 8-character hexadecimal string.

host USB host as defined in [USB]

initiator PTP initiator as defined in [PTP]

operation Generally, a computing function and associated data.

responder PTP responder as defined in [PTP]

UI User Interface. Features that enable the end user to interact with a device, e.g., buttons, keyboards, LCD displays.

5. Architectural Framework

The DPS architecture is designed to operate at the application layer and to be independent of the specifics of the underlying data transport. It defines mechanisms for:

- Discovering device capabilities
- Job initiation and configuration
- Retrieval of print data by the Printer
- Asynchronous notification to the DSC by the Printer of key events and status changes

DPS actions take the form of **operations** and **events**. Asynchronous events are sent from the Printer to the DSC.

DPS is also designed around a flexible system architecture, where system functional components are not rigidly assigned to a particular device type. The general DPS model is described in section 5.1. However, this document—DPS Version 1.1—specifies a more limited and restricted version of the general architecture. These limitations and restrictions are described in section 5.2.

5.1 General DPS Model

The DPS architecture employs a simple Client / Server model.

5.1.1 System Components

Print Client—controls the flow of system operations. Requests the print job, indicating which images should be printed and what attributes should be applied to them; monitors printing progress and reports status to the user. This component is also responsible for gathering input from the user about the print request.

Print Server—controls the physical print engine and manages all necessary data flow to and from the Printer. It receives print requests, fetches images from storage and reports both normal status and error conditions to the Print Client.

Storage Server—manages access to physical storage. It provides an abstraction that is physical device and file system independent. The general DPS system allows multiple physical stores distributed among more than one physical device.

Storage Client-—retrieves print data from the Storage Server for processing by the print engine.

The general DPS system model may be viewed as follows:

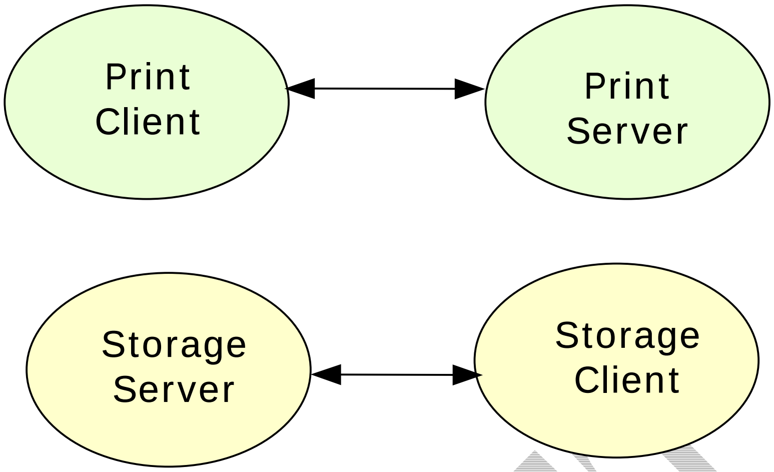

Figure 5.1-1 General DPS System Model

For the general DPS system architecture, all components could be located in different physical devices.

5.2 DPS Architecture for Version 1.1

5.2.1 Devices and Transports Supported

DPS Version 1.1 is defined to support two physical devices—a DSC and a Printer—over one physical connection—USB. The Printer is the USB host. The DSC is a USB Still Image Capture Device (SICD). As required by [SICD], the DSC and the Printer support the Picture Transfer Protocol (PTP) as defined in [PTP].

See Appendix B for the details of how DPS works over PTP/USB.

5.2.2 System Components

The following four system components are defined for Version 1.1.

In the DSC:

Print Client— controls the flow of system operations after the configuration step is completed. Requests the print job, indicating which images should be printed and what attributes should be applied to them; monitors printing progress and reports status to the user. This component is also responsible for gathering input from the user about the print request.

Storage Server—manages access to the DSC's physical storage. It provides an abstraction that is physical device and file system independent.

In the Printer:

Print Server— processes requests for print jobs, printer capabilities, status, etc. Provides asynchronous notification of progress and status. Manages overall printer resources, including print engine.

Storage Client—retrieves objects including images from the DSC's Storage Server.

The Version 1.1 architecture can be represented as shown in Figure 5.2-1.

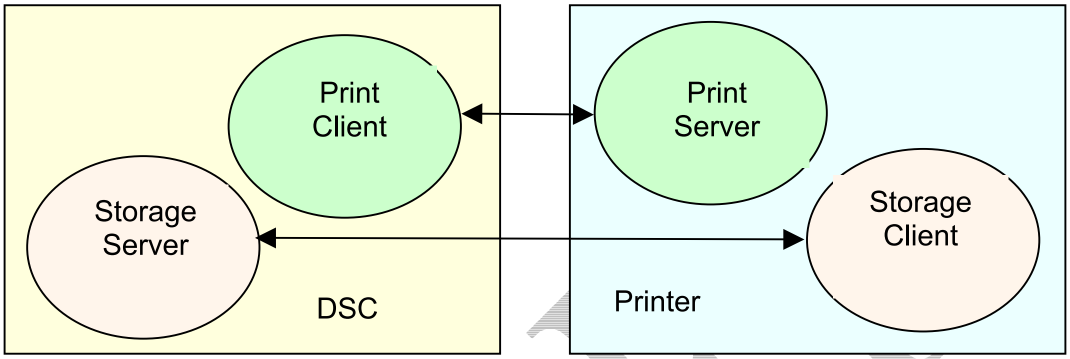

Figure 5.2-1 DPS System Model Version 1.1

5.2.3 Design Constraints

Consistent with the design objective to require a very small memory footprint, especially on the DSC, this version of DPS is designed to enable a DSC to implement all defined user capabilities with a one kilobyte input buffer. That is, all requests from the DSC may be structured such that the response from the Printer will contain no more than 1024 bytes. At least a 64-kilobyte input buffer is required of the Printer.

5.2.4 System Requirements

See Appendix B and Appendix D.

5.3 Device and Service Discovery

As device and service discovery processes are specific to a physical link and / or the operating environment, and since DPS is an application layer protocol that is independent of the underlying transport, DPS does not define the mechanism by which the devices or services discover each other.

The appendix relevant to the underlying link in question shall supply the details of the discovery process for DPS applications on that link. For PTP/ USB, see Appendix B.

5.4 DPS Job Overview

As discussed in section 5.3, the actual device discovery process is link dependent. After discovery, if the job is controlled from the DSC, the Print Client in the DSC queries the Printer about its capabilities and then, issues a request for a print job conforming to the

Printer's capabilities. The Printer retrieves the print data from the DSC's image Storage Server, and provides status information, including an indication that the job has completed. If the job is controlled from the Printer, the Printer simply retrieves the appropriate print data from the DSC and prints it, providing the user with ongoing status.

5.5 Example of DPS Job Flow

Figure 5.5-1shows a sequence diagram providing a conceptual model of the flow of DPS operations, where the Print Client and Storage Server reside on the DSC. (The flow of responses is not shown.)

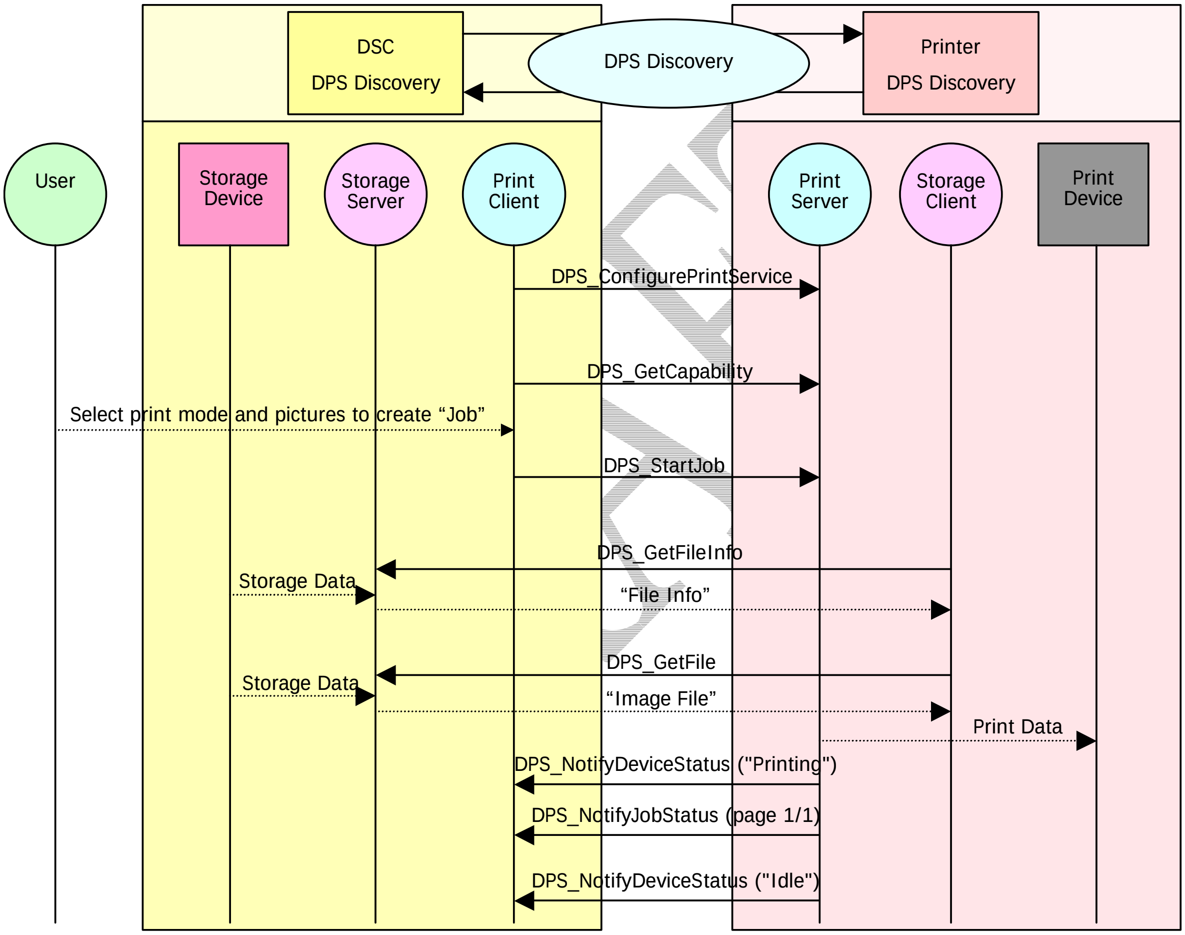

Figure 5.5-1 DPS Job Flow Example

6. DPS Actions: Operations and Events

6.1 DPS Actions Overview

DPS has two kinds of actions: DPS operations and DPS events. See section 6.2 and section 6.3 for each definition. DPS actions consist of a request phase followed by a response phase; both the request and response are well formed XML. The request includes 1 or more input parameters; the response includes 1 or more output parameters, where the first output parameter is the result of the operation. All DPS operations and responses are considered synchronous. Only one request may be outstanding at any point in time; that is, a request shall be followed by its response before the next request may be issued.

XML markup contents and attribute values utilize ISO-8859-1 (Latin 1) character repertoires. See [LatinChars] for details. All XML script examples in this document are assumed to be UTF-8 encoded.

6.1.1 DPS Actions Description Contents

The description of each operation and event specifies:

- Required and optional parameters and their semantics
- Error reporting
- Conformance requirements for both Printer and DSC
- An example XML script implementing the operation

Specifics of the syntax are defined by the XML Schema. See [XML Schema] and Appendix A for more information. If the operation, event, or parameter description and the schema do not agree, the schema takes precedence.

6.2 DPS Operations

A DPS operation is a transaction initiated by the DPS client and completed by the DPS server. It consists of two phases: first a request is sent from the DPS client to the DPS server; then a response is sent from the DPS server to the DPS client.

To invoke a DPS operation, a client shall send an XML script in the following format. Values in italics are placeholders for actual values.

```
<?xml version=”1.0”?>
<dps xmlns=”http://www.cipa.jp/dps/schema/”>
 <input>
  <operationName>
   <argumentName>in argument value</argumentName>
   Other in arguments and their values go here, if any.
  </operationName>
 </input>
```

```
</dps>
```

A dps element shall have only one input element.

An input element shall have only one DPS operation element.

All header values, element names, and element values are case-sensitive except where noted.

6.3 DPS Events

A DPS event is a notification from a DPS server to the DPS client. It is initiated when a state change occurs in the server. Events are sent asynchronously to the flow of operation requests and responses. Events also consist of a request phase followed by a response phase.

To communicate a DPS event, a server shall send an XML script in the following format. Values in italics are placeholders for actual values.

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <eventName>
   <argumentName>in argument value</argumentName>
   Other in arguments and their values go here, if any.
  </eventName>
 </input>
</dps>
```

A dps element shall have only one input element.

An input element shall have only one DPS event element.

All header values, element names, and element values are case-sensitive except where noted.

6.4 DPS Action Results

For every DPS Action request, a DPS Action result shall be returned along with the DPS Action response. Table 6.4-1 lists the DPS Action results and their descriptions. See section 10.3 for more information.

DPS Action Result	
Result	**Description**
OK	The request was successfully processed.
Not executed	The request was understood, but was not executed.
Operation not supported	A request was received and was understood, but was not supported.
Operation not recognized	A request was received, but the operation or event requested was not recognized.

Table 6.4-1 DPS Action Results

A DPS result shall be sent in the following format. Values in italics are placeholders for actual values.

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>One of the values in Table 10.3-1</result>
  <operationName>
   <argumentName>in argument value</argumentName>
   Other in arguments and their values go here, if any.
  </operationName>
 </output>
</dps>
```

A dps element shall have only one output element.

An output element shall have one result element and, if the request was understood, one DPS operation element.

All header values, element names, and element values are case-sensitive except where noted.

If the receiver cannot recognize the operation, the result shall be sent in the following format.

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
```

```
  <result>********</result>
 </output>
</dps>
```

A dps element shall have only one output element.

An output element shall have only one result element.

All header values, element names, and element values are case-sensitive except where noted.

6.5 Collision Situation

Each DPS Action consists of a request and a response. These may be implemented as a sequence of lower-level transactions specific to the underlying transport. The transport might not arbitrate these DPS Actions at the transaction level. That could lead to collisions.

Figure 6.5-1 shows an example of a collision. Printer sends DPS Request A and DSC sends DPS Request B simultaneously. In this case, Printer receives DPS Request B while it expects DPS Response A. On the other hand, DSC receives DPS Request A instead of DPS Response B. These situations are defined as collision.

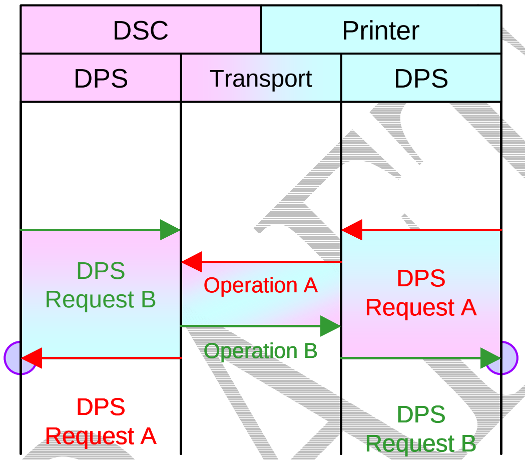

Figure 6.5-1 Collision situation

There may be transport-dependent request/response collision problems. See Appendix B.7 details regarding PTP / USB.

7. DPS Print Service

The DPS Print Server provides the DPS Print Service. The DPS Print Server processes DPS operation requests from the Print Client and provides event notifications to the Print Client.

7.1 Print Service Operations and Events

A Printer providing the DPS Print Service accepts requests from clients for the following operations:

Operation Name	Description
DPS_ConfigurePrintService	Exchange versions and other configurations to be used.
DPS_GetCapability	Obtain information about the Printer's capabilities.
DPS_GetJobStatus	Get job status from the Printer.
DPS_GetDeviceStatus	Get device status from the Printer.
DPS_StartJob	Establish configuration for and initiate a print job.
DPS_AbortJob	Abort all DPS print jobs.
DPS_ContinueJob	Resume a print job after a pause.

Table 7.1-1 Print Service Operations

In addition, a Printer providing the DPS Print Service provides the following event notifications:

Event Name	Description
DPS_NotifyJobStatus	Notify the client of a change in print job status.
DPS_NotifyDeviceStatus	Notify the client of a change in device status.

Table 7.1-2 Print Service Events

7.2 Print Service Operations - Detailed Descriptions

7.2.1 DPS_ConfigurePrintService

Parameters for DPS_ConfigurePrintService

	Parameter		Description
Input	configurePrintService		Parameters for DPS_ConfigurePrintService request.
		dpsVersions	The list of DPS Versions that can be supported by the Print Client. Each version number shall be represented by a decimal string of no more than 8 characters (8 digits or 7 digits and a decimal point).
		vendorName	DSC manufacturer's name.
		vendorSpecificVersion	The vendor-specific profile version supported by the Print Client.
		productName	Product name of DSC.
		serialNo	The device serial number of the DSC, provided as an ASCII string.
Output	result		Operation result.
	configurePrintService		Parameters for DPS_ConfigurePrintService response.
		printServiceAvailable	Indicates whether or not the Print Server is available for processing the Print Client's job.
		dpsVersions	The list of DPS Versions that shall be used for this print service.
		vendorName	Printer manufacturer's name.
		vendorSpecificVersion	The vendor-specific profile version supported by the Print Server.
		productName	Product name of Printer.
		serialNo	Device serial number of the Printer, provided as an ASCII string.

Table 7.2-1 Parameters for DPS_ConfigurePrintService

See section 10.4 for details regarding DPS_ConfigurePrintService parameters.

The Print Client indicates to the Print Server in the request the list of DPS Versions they are capable of using, along with its device information. The Print Server responds with the list of DPS Versions it can support.

All DPS-compliant devices shall support DPS Version 1.0.

DPS Version1.1 compliant devices shall also support DPS Version 1.1.

In the list of DPS Versions, all published DPS Versions shall be enumerated in ascending order. Server and Client shall use maximum common version.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <configurePrintService>
   <dpsVersions>1.0 1.1</dpsVersions>
   <vendorName>DSC_Vendor</vendorName>
   <vendorSpecificVersion>1.0</vendorSpecificVersion>
   <productName>DPS_DSC</productName>
   <serialNo>ABC01234567</serialNo>
  </configurePrintService>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <configurePrintService>
   <printServiceAvailable>********</printServiceAvailable>
   <dpsVersions>1.0 1.1</dpsVersions>
   <vendorName>Printer_Vendor</vendorName>
   <vendorSpecificVersion>1.0</vendorSpecificVersion>
   <productName>DPS_Printer</productName>
   <serialNo>ABA987654321</serialNo>
  </configurePrintService>
 </output>
</dps>
```

7.2.2 DPS_GetCapability

Parameters for DPS_GetCapability

	Parameter				Description
Input	getCapability				Parameters for DPS_GetCapability request.
		capability			Parameter specifying the printer attribute being requested.
			qualities		Requests a space-separated list of quality levels supported by the Printer.
			paperSizes		Requests a space-separated list of paperSize values supported by the Printer.
			paperTypes		Requests a space-separated list of the paperType values supported by the Printer. When unqualified by the paperSize attribute, the entire set of paperTypes supported by the Printer is requested. When qualified by the paperSize attribute, only the set of paperTypes supported for the indicated paperSize is returned.
				paperSize	When the paperTypes parameter is qualified by a paperSize attribute, only the list of paperType values supported for the specified paperSize is requested.
			fileTypes		Requests a space-separated list of fileType values supported by the Printer.
			datePrints		Requests information as to whether the Printer supports printing dates adjacent to images.
			fileNamePrints		Requests information as to whether the Printer supports printing file names adjacent to images.
			imageOptimizes		Requests information as to whether the Printer can enhance or "optimize" images.
			layouts		Request for information about the Printer's ability to support various page layouts. This parameter shall contain an attribute specifying the paperSize for which layout information is requested. That is, this parameter requests a list of layout parameters supported for a particular paperSize.
				paperSize	Attribute for the layouts parameter indicating the paperSize for which layout information is being requested.
			fixedSizes		Requests a list of values supported by the Printer for fixed image size output.
			croppings		Requests an indication as to whether the Printer supports the cropping of images.
			charRepertoires V1.1		Requests a space-separated list of the sets of characters that the Printer is capable of printing when processing an XHTML-Print file.
Output	result				Operation result.
	getCapability				Parameters for DPS_GetCapability response.

		capability	Parameter corresponding to input capability parameter. The list of values supported by the Print Server for the parameter is returned.

Table 7.2-2 Parameters for DPS_GetCapability

See section 10.5 for details regarding DPS_GetCapability parameters.

The DPS_GetCapability operation enables the DSC to learn about the capabilities of the Printer. The request form specifies a capability parameter with a single child parameter and possibly an associated attribute. The response takes the same form and provides information as to the specifics of the Printer's capabilities regarding the requested parameter.

When the request is for supported `paperTypes`, the `paperSize` attribute may be provided, and the types supported for that particular media size are returned. If the `paperSize` attribute is not provided in the request, the response shall contain all `paperTypes` supported by the Printer for all `paperSizes`.

When the request is for the set of supported layouts, the `paperSize` attribute shall be provided, and the set of supported `layouts` for that `paperSize` is returned.

For a definition of the value set appropriate to each parameter, see section 10.5.

Execution Examples:

Input Example 1:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getCapability>
   <capability>
    <datePrints/>
   </capability>
  </getCapability>
 </input>
</dps>
```

Output Example 1:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getCapability>
   <capability>
    <datePrints>******** ********</datePrints>
   </capability>
  </getCapability>
 </output>
</dps>
```

Input Example 2:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getCapability>
   <capability>
    <paperTypes paperSize="********"/>
   </capability>
  </getCapability>
 </input>
</dps>
```

Output Example 2:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getCapability>
   <capability>
    <paperTypes paperSize="********">
     ******** ******** ********
    </paperTypes>
   </capability>
  </getCapability>
 </output>
</dps>
```

Input Example 3:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getCapability>
   <capability>
    <layouts paperSize="********"/>
   </capability>
  </getCapability>
 </input>
</dps>
```

Output Example 3:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getCapability>
   <capability>
    <layouts paperSize="********">
  ******** ******** ******** ********
    </layouts>
   </capability>
  </getCapability>
 </output>
</dps>
```

7.2.3 DPS_GetJobStatus

Parameters for DPS_GetJobStatus

	Parameter	Description
Input	getJobStatus	Parameters for DPS_GetJobStatus request.
Output	result	Operation result.
	getJobStatus	Parameters for DPS_GetJobStatus response.
	prtPID	Three-digit decimal job number defined in the DPOF file. *Only present when the Printer is processing a DPOF file; a response shall not contain both* prtPID *and* imagesPrinted*.*
	filePath	Path for the currently printing image data file, which is described in the DPOF file; or, when more than one image is printed on a single page of media, the path for the initial image file described in the DPOF file. *Only present when the Printer is processing a DPOF file; a response shall not contain both* filePath *and* imagesPrinted*.*
	copyID	Three-digit decimal value that represents the current copy number of the DPOF print job that is identified by prtPID. *Only present when the Printer is processing a DPOF file; a response shall not contain both* copyID *and* imagesPrinted*.*
	progress	String value of *N*/*T* where *N* is the three-digit decimal number of the current page being printed and *T* is the three-digit decimal total number of pages to be printed for this job. The minimum value of *N* and *T* is one (001). If multiple jobs have been submitted by the DSC, the progress reported is for the first job submitted that has not yet completed. *When the Printer is processing an XHTML-Print file and if the total number of pages cannot be estimated, T is set to 999.*
	imagesPrinted	Three-digit decimal value indicating the number of images in the current job that have been printed. In the case of multiple copies of an image, each copy of the image shall be included in the overall count. This parameter shall be updated at the beginning of each page. *A response shall not contain both* imagesPrinted *and* prtPID *or* filePath *or* copyID*.* *When the Printer is processing an XHTML-Print file, a response shall not contain* imagesPrinted*.*

Table 7.2-3 Parameters for DPS_GetJobStatus

See section 10.6 for details regarding DPS_GetJobStatus parameters.

The DSC's Print Client calls DPS_GetJobStatus to get the Printer's current DPS job status. When more than one job has been submitted, the status information returned is relative to the first DPS job submitted that has not yet completed.

If the job for which status is returned is a DPOF job, the response shall include the `progress` parameter and may include `prtPID`, `filePath` and `copyID` parameters; it shall not include the `imagesPrinted` parameter. If the Printer does not support the DPOF restart capability (see Appendix C.1.4), it shall not include the `prtPID`, filePath and `copyID` parameters in the DPS_GetJobStatus response or a DPS_NotifyJobStatus request.

If the job is an XHTML-Print job, the response shall include only the `progress` parameter. Other than a DPOF or an XHTML-Print job, the response shall contain the `progress` and `imagesPrinted` parameters, and shall not contain the `prtPID`, `filePath` or `copyID` parameters.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getJobStatus/>
 </input>
</dps>
```

Output Example 1:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getJobStatus>
   <progress>004/006</progress>
   <imagesPrinted>012</imagesPrinted>
  </getJobStatus>
 </output>
</dps>
```

Output Example 2:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getJobStatus>
   <prtPID>001</prtPID>
   <filePath>../DCIM/100ABCDE/FGHI0001.JPG</filePath>
   <copyID>002</copyID>
   <progress>004/006</progress>
  </getJobStatus>
 </output>
```

```
</dps>
```

7.2.4 DPS_GetDeviceStatus

Parameters for DPS_GetDeviceStatus

	Parameter		Description
Input	getDeviceStatus		Parameters for DPS_GetDeviceStatus request.
Output	result		Operation result.
	getDeviceStatus		Parameters for DPS_GetDeviceStatus response.
		dpsPrintServiceStatus	Current print service status.
		jobEndReason	Indicates whether or not a job has terminated, and if so, whether it terminated normally or abnormally.
		errorStatus	Current Printer status, including ‘no error’.
		errorReason	Reason(s) for current status, if other than ‘no error’.
		disconnectEnable	Indicates whether or not it is OK for the DSC to disconnect, since the Printer has all data necessary for the print job.
		capabilityChanged	Indicates whether or not the Printer's capabilities have changed since the last DPS_GetCapability operation.
		newJobOK	Indicates that the Printer is ready to receive another job from the DSC.

Table 7.2-4 Parameters for DPS_GetDeviceStatus

See section 10.7 for defined values for DPS_GetDeviceStatus.

The Print Client on the DSC sends DPS_GetDeviceStatus to get the Printer’s current device status.

See section 11, for event transitions of the following parameters: dpsPrintServiceStatus, jobEndReason, errorStatus, errorReason, disconnectEnable, capabilityChanged, and newJobOK.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getDeviceStatus/>
 </input>
```

```
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getDeviceStatus>
   <dpsPrintServiceStatus>********</dpsPrintServiceStatus>
   <jobEndReason>********</jobEndReason>
   <errorStatus>********</errorStatus>
   <errorReason>********</errorReason>
   <disconnectEnable>********</disconnectEnable>
   <capabilityChanged>********</capabilityChanged>
   <newJobOK>********</newJobOK>
  </getDeviceStatus>
 </output>
</dps>
```

7.2.5 DPS_StartJob

Parameters for DPS_StartJob

	Parameter			Description
Input	startJob			Parameters for DPS_StartJob request.
		jobConfig		The requested job configuration parameters.
			quality	Print quality level selected for the job.
			paperType	Paper type selected for the job.
			paperSize	Paper size selected for the job.
			fileType	Image type for all the images selected for the job.
			datePrint	Whether or not to print dates provided by printInfo date.
			fileNamePrint	Whether or not to print file names given in printInfo fileName.
			imageOptimize	Whether or not to enhance or optimize images.
			layout	The page layout selected for the job.
			fixedSize	If present, provides the physical dimensions of the output image size for all images in this job. Note that fixedSize shall not be specified with layout at the same time.
			cropping	Whether or not to specify the cropping area to be printed.
		printInfo		Parameters for each image to be printed.
			croppingArea	Position of upper-left, width and height of the area to be printed (See section 10.8.1 for details).
			fileID	DPS fileID for image to be printed.
			fileName	Filename to be printed.
			date	Date to be printed adjacent to image given by the Print Client (on DSC). The date is given as an arbitrary string.
			copies	Number of copies to be printed of the image corresponding to fileID. Number of copies to be printed of the XHTML-Print document referenced by filePath, when the printer is processing the XHTML-Print file.
			prtPID	prtPID is used for the DPOF restart operation. See Appendix C.1.4. Also see sections 7.2.3 and 7.3.1.
			filePath	filePath is used for the DPOF restart operation and the XHTML-Print operation. For the DPOF restart operation, see Appendix C.1.4. Also see sections 7.2.3 and 7.3.1. For the XHTML-Print operation, see Appendix H.1.2.
			copyID	copyID is used for the DPOF restart operation. See Appendix C.1.4. Also see sections 7.2.3 and 7.3.1.
Output	result			Operation result.
	startJob			Parameters for DPS_StartJob response.

Table 7.2-5 Parameters for DPS_StartJob

See section 10.8 for defined values for DPS_StartJob.

DPS_StartJob initiates a print job by sending a complete job description.

jobConfig is used to configure print settings to print the images described by the following one or more occurrences of printInfo. The printInfo parameter may be repeated multiple times except as described for DPOF jobs and XHTML-Print jobs below. If fixed-size prints are requested, and the source image aspect ratio does not match the aspect ratio of the desired fixed size, the Print Server should scale and crop the source data as necessary.

If the DPS_StartJob is a request for the Print Server to directly process a DPOF file, the following special conditions apply:

- No other objects may be combined with the DPOF request; the DPOF file shall be the only file submitted for the job, that is, only one printInfo parameter and only one fileID may appear in the DPS_StartJob request.
- If the configuration provided with this operation is in conflict with attributes provided by the DPOF file, the DPOF file shall take precedence.

Concrete examples are described in Appendix C.

If the DPS_StartJob is a request for the Print Server to process an XHTML-Print file, the following special conditions apply:

- No other objects may be combined with the XHTML-Print request; the XHTML-Print file shall be the only file submitted for the job, that is, only one printInfo parameter may appear in the DPS_StartJob request.
- The quality, paperType, paperSize, fileType and imageOptimize parameters in jobConfig and the fileID, copies and filePath parameters in printInfo are valid. If parameters other than these are set, the Print Server ignores them.
- The fileType parameter in jobConfig shall be set to XHTML-Print (********). And the fileID parameter in printInfo is set to ********.
- The copies parameter in printInfo indicates the number of copies to be printed of the XHTML-Print document. The DSC shall retain the XHTML-Print file and referenced files for the Printer to access the files for every copy until the print job is finished or disconnectEnable is set to True.
- The filePath parameter in printInfo shall be the absolute URI of the XHTML-Print source document.

Concrete examples are described in Appendix H.

Execution Examples:

Input Example 1:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
```

```
 <input>
  <startJob>
   <jobConfig>
    <quality>********</quality>
    <paperSize>********</paperSize>
    <paperType>********</paperType>
    <fileType>********</fileType>
    <datePrint>********</datePrint>
    <fileNamePrint>********</fileNamePrint>
    <imageOptimize>********</imageOptimize>
    <layout>********</layout>
    <cropping>********</cropping>
   </jobConfig>
   <printInfo>
    <fileID>00000001</fileID>
    <fileName>Test0001</fileName>
    <date>2002/10/28</date>
   </printInfo>
   <printInfo>
    <fileID>00000002</fileID>
    <fileName>Test0002</fileName>
    <date>31 Oct, 2002</date>
    <croppingArea>0000 0000 0280 01E0</croppingArea>
    <copies>002</copies>
   </printInfo>
  </startJob>
 </input>
</dps>
```

Input Example 2 (fixed-size printing):

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <startJob>
   <jobConfig>
    <quality>********</quality>
    <paperSize>********</paperSize>
    <paperType>********</paperType>
    <fileType>********</fileType>
    <datePrint>********</datePrint>
    <fileNamePrint>********</fileNamePrint>
    <imageOptimize>********</imageOptimize>
    <fixedSize>********</fixedSize>
    <cropping>********</cropping>
   </jobConfig>
   <printInfo>
    <fileID>00000001</fileID>
    <date>2002/10/28</date>
   </printInfo>
  </startJob>
 </input>
</dps>
```

7.2.6 DPS_AbortJob

Parameters for DPS_AbortJob

	Parameter		Description
Input	abortJob		Parameters for DPS_AbortJob request.
		abortStyle	Indicates whether the Print Server should abort the DSC's job immediately or after printing to the end of the current page.
Output	result		Operation result.
	abortJob		Parameters for DPS_AbortJob response.

Table 7.2-6 Parameters for DPS_AbortJob

See section 10.9 for defined values for DPS_AbortJob.

DPS_AbortJob sends an abort job request to the Printer. The Printer shall terminate all DPS print job at the earliest opportunity upon receipt of the DPS_AbortJob request.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <abortJob>
   <abortStyle>********</abortStyle>
  </abortJob>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <abortJob/>
 </output>
</dps>
```

7.2.7 DPS_ContinueJob

Parameters for DPS_ContinueJob

	Parameter	Description
Input	continueJob	Parameters for DPS_ContinueJob request.
Output	result	Operation result.
	continueJob	Parameters for DPS_ContinueJob response.

Table 7.2-7 Parameters for DPS_ContinueJob

DPS_ContinueJob sends a resume printing command. If the Printer has paused for some reason, this is a request to resume, if possible.

For example, if the Print Server experiences an error such as running out of paper in the middle of the job, the Print Client may wish to present the user with a prompt screen providing choices including "continue" and "cancel". After refilling the media tray, the user may select "continue". The DSC would then issue DPS_ContinueJob, requesting the Print Server to try to continue job processing.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <continueJob/>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <continueJob/>
 </output>
</dps>
```

7.3 Print Service Events - Detailed Descriptions

7.3.1 DPS_NotifyJobStatus

Parameters for DPS_NotifyJobStatus

	Parameter	Description
Input	`notifyJobStatus`	Parameters for DPS_NotifyJobStatus request.
	`prtPID`	Job number defined in the DPOF file. *Only present when the Printer is processing a DPOF file; a request shall not contain both* `prtPID` *and* `imagesPrinted`*.*
	`filePath`	Path for the currently printing image data file, which is described in the DPOF file; or, when more than one image is printed on a single page of media, the path for the initial image file described in the DPOF file. *Only present when the Printer is processing a DPOF file; a request shall not contain both* `filePath` *and* `imagesPrinted`*.*
	`copyID`	Three-digit decimal value that represents current copy number of the DPOF print job that is identified by `prtPID`. *Only present when the Printer is processing a DPOF file; a request shall not contain both* `copyID` *and* `imagesPrinted`*.*
	`progress`	String value of *N*/*T* where *N* is the number of the current page being printed and *T* is the total number of pages to be printed for this job. The minimum value of *N* and *T* is one (001). If multiple jobs are queued in the Printer, the progress reported corresponds to the first job submitted that has not yet completed. *When the Printer is processing an XHTML-Print file and if the total number of pages cannot be estimated, T is set to 999.*
	`imagesPrinted`	Three-digit decimal value indicating how many images have been printed for the current job. In the case of multiple copies of an image, each copy of the image shall be included in the overall count. This parameter shall be updated at the beginning of each page. This parameter shall not be specified with prtPID, filePath or copyID at the same time. *When the Printer is processing an XHTML-Print file, this parameter shall not be specified.*
Output	`result`	Event result.
	`notifyJobStatus`	Parameters for DPS_NotifyJobStatus response.

Table 7.3-1 Parameters for DPS_NotifyJobStatus

See section 10.6 for defined values for DPS_NotifyJobStatus parameters.

DPS_NotifyJobStatus updates the DSC if there are changes in the status of a job submitted by the DSC. The Printer Server should issue DPS_NotifyJobStatus within a reasonable time.

The Print Server shall issue DPS_NotifyJobStatus upon beginning to print each page of the job. See section 11 for a summary of event transitions.

Execution Example:

Input Example 1:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <notifyJobStatus>
   <prtPID>001</prtPID>
   <filePath>../DCIM/100ABCDE/FGHI0001.JPG</filePath>
   <copyID>002</copyID>
   <progress>004/006</progress>
  </notifyJobStatus>
 </input>
</dps>
```

Input Example 2:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
   <notifyJobStatus>
   <progress>004/006</progress>
   <imagesPrinted>012</imagesPrinted>
  </notifyJobStatus>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <notifyJobStatus/>
 </output>
</dps>
```

7.3.2 DPS_NotifyDeviceStatus

Parameters for DPS_NotifyDeviceStatus

	Parameter		Description
Input	notifyDeviceStatus		Parameters for DPS_NotifyDeviceStatus request.
		dpsPrintServiceStatus	Current print service status.
		jobEndReason	Indicates whether the job terminated normally or abnormally.
		errorStatus	Current Printer status, including ‘no error’.
		errorReason	Reason(s) for current status, if other than ‘no error’.
		disconnectEnable	Indicates whether or not it is OK for the DSC to disconnect, since the Printer has all data necessary for the print job.
		capabilityChanged	Indicates whether or not the Printer’s capabilities have changed since the last DPS_GetCapability operation.
		newJobOK	Indicates that the Printer is ready to receive another job from the DSC.
Output	result		Event result.
	notifyDeviceStatus		Parameters for DPS_NotifyDeviceStatus response.

Table 7.3-2 Parameters for DPS_NotifyDeviceStatus

See section 10.7 for defined values for DPS_NotifyDeviceStatus.

The Print Server issues a DPS_NotifyDeviceStatus to notify the DSC of changes in the Printer's device status. The Printer Server should issue DPS_NotifyDeviceStatus within a reasonable time.

If disconnectEnable indicates that data transfer is complete, the DSC may disconnect to free the DSC (and the user) or may stay connected to obtain job progress information. DPS_NotifyDeviceStatus shall be issued by the Print Server upon a change in value of any of the input parameters.

See section 11 for event transitions of the following parameters: `dpsPrintServiceStatus`, `jobEndReason`, `errorStatus`, `errorReason`, `disconnectEnable`, `capabilityChanged` and `newJobOK`.

Execution Example:

Input Example:

```
   <?xml version="1.0"?>
```

```
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <notifyDeviceStatus>
   <dpsPrintServiceStatus>********</dpsPrintServiceStatus>
   <jobEndReason>********</jobEndReason>
   <errorStatus>********</errorStatus>
   <errorReason>********</errorReason>
   <disconnectEnable>********</disconnectEnable>
   <capabilityChanged>********</capabilityChanged>
   <newJobOK>********</newJobOK>
  </notifyDeviceStatus>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <notifyDeviceStatus/>
 </output>
</dps>
```

8. DPS Storage Service

The DPS Storage Server provides the DPS Storage Service. The DPS Storage Server processes DPS operation requests from the Storage Client. There are no events defined for the Storage Service.

For Version 1.1, most operations defined in this section are informative only, and shall not actually be transmitted between the DSC and the Printer. The actual retrieval of print data is out of band to DPS Version 1.1 (this version), and shall be implemented by utilizing PTP operations rather than the DPS operations defined in this section. See Appendix B.6.3 for details.

8.1 Storage Service Operations

A storage device providing the DPS Storage Service accepts requests from clients for the following operations:

Operation Name	Description
DPS_GetFileID	Get file identifier.
DPS_GetFileInfo	Get file information.
DPS_GetFile	Get file from storage.
DPS_GetPartialFile	Get partial file from storage.
DPS_GetFileList	Get file list from storage.
DPS_GetThumb	Get thumbnail data.

Table 8.1-1 Storage Service Operations

8.2 Storage Service Operations – Detailed Descriptions

8.2.1 DPS_GetFileID

Parameters for DPS_GetFileID

	Parameters		Description
Input	getFileID		Parameters for DPS_GetFileID request.
		basePathID	The fileID of the base directory for the filePath parameter. The fileID of a file can be specified as basePathID. When the printer is processing a DPOF file, the basePathID shall be the object ID for the "AUTPRINT.MRK" file. When the printer is processing an XHTML-Print file, the basePathID shall be set to ********.
		filePath	The path name of the file whose ID is to be returned. This path is relative to the base directory given by basePathID. When the printer is processing an XHTML-Print file, this path shall be the absolute URI.
Output	result		Operation result.
	getFileID		Parameters for DPS_GetFileID response.
		fileID	The DPS fileID of the image referred to by filePath.

Table 8.2-1 Parameters for DPS_GetFileID

See section 10.10 for defined values for DPS_GetFileID.

When the DSC is executing DPOF AUTPRINT Images Print or XHTML-Print, it shall support DPS_GetFileID. Since the DPS Storage Service makes no assumptions about a file system structure, the filePath from a file such as a DPOF file or an XHTML-Print file cannot be used directly. DPS_GetFileID requests the storage server to convert a file path for DPOF AUTPRINT Images Print or a URI for XHTML-Print to a fileID that can be used by the DPS storage client. The fileID shall remain valid for the duration of the current DPS session, as required for all DPS fileIDs.

When the filePath is set to an absolute URI that has the pictbridge scheme, the basePathID parameter shall be set to ********, and the DSC shall return the value of the corresponding fileID to the Printer.

Execution Examples:

Input Example 1:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getFileID>
   <basePathID>00000002</basePathID>
   <filePath>../DCIM/100ABCDE/FGHI0001.JPG</filePath>
  </getFileID>
 </input>
</dps>
```

Input Example 2:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getFileID>
   <basePathID>********</basePathID>
   <filePath>pictbridge://storageserver/DCIM/100ABCDE/FGHI0001.JPG
   </filePath>
  </getFileID>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getFileID>
   <fileID>00000003</fileID>
  </getFileID>
 </output>
</dps>
```

8.2.2 DPS_GetFileInfo (Informative)

Parameters for DPS_GetFileInfo

	Parameters		Description
Input	getFileInfo		Parameters for DPS_GetFileInfo request.
		fileID	Identifier of the file to get information about.
Output	result		Operation result.
	getFileInfo		Parameters for DPS_GetFileInfo response.
		fileType	Type of file referenced by fileID.
		fileSize	Size, in bytes, of the file referenced by fileID (8-character hexadecimal).
		thumbFormat	Indicates whether the thumbnail format is compressed or uncompressed.
		thumbSize	Size in bytes of the thumbnail file associated with fileID (8-character hexadecimal).

Table 8.2-2 Parameters for DPS_GetFileInfo

See section 10.11 for defined values for DPS_GetFileInfo.

DPS_GetFileInfo enables the Printer to learn about the attributes of a file before actually retrieving it to print.

Implementer's Note: For DPS Version 1.1, this operation will be directly mapped to a transport-dependent operation. See Appendix B.6.3.1.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getFileInfo>
   <fileID>00000001</fileID>
  </getFileInfo>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
```

```
 <output>
  <result>********</result>
  <getFileInfo>
   <fileType>********</fileType>
   <fileSize>000F4240</fileSize>
   <thumbFormat>********</thumbFormat>
   <thumbSize>00003A43</thumbSize>
  </getFileInfo>
 </output>
</dps>
```

8.2.3 DPS_GetFile (Informative)

Parameters for DPS_GetFile

	Parameters		Description
Input	getFile		Parameters for DPS GetFile request.
		fileID	Identifier of the file to read.
Output	result		Operation result.
	getFile		Parameters for DPS_GetFile response.
		bytesRead	Actual size in bytes of file that was transferred (8-character hexadecimal).

Table 8.2-3 Parameters for DPS_GetFile

See section 10.12 for defined values for DPS_GetFile.

This operation transfers the actual print data (e.g., a JPEG file) from the Storage Server on the DSC to the Storage Client on the Printer.

Implementer's Note: For DPS Version 1.1, this operation will be directly mapped to a transport-dependent operation. See Appendix B.6.3.2.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getFile>
   <fileID>00000001</fileID>
  </getFile>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getFile>
   <bytesRead>000E848A</bytesRead>
  </getFile>
 </output>
```

```
</dps>
```

8.2.4 DPS_GetPartialFile (Informative)

Parameters for DPS_GetPartialFile

	Parameters	Description
Input	`getPartialFile`	Parameters for DPS GetPartialFile request.
	`fileID`	Identifier of the file to read.
	`offset`	The offset in bytes from the beginning of the referenced file to the position from which to start reading data (8-character hexadecimal).
	`maxSize`	Maximum number of bytes to read from the file (8-character hexadecimal).
Output	`result`	Operation result.
	`getPartialFile`	Parameters for DPS_GetPartialFile response.
	`bytesRead`	Number of bytes transferred. May be less than the number requested (8-character hexadecimal).

Table 8.2-4 Parameters for DPS_GetPartialFile

See section 10.13 for defined values for DPS_GetPartialFile.

DPS_GetPartialFile is used to read part of a data file when the Printer either doesn't have room to store the whole file or it doesn't need the whole file. The `maxSize` parameter indicates the maximum number of bytes to read from the file. If end-of-file is reached before reading `maxSize` bytes from the file, the transfer is terminated and the output parameter indicates how many bytes were actually transferred. If `bytesRead` is less than `maxSize`, it indicates that end-of-file was reached.

Note that even if `bytesRead` is equal to `maxSize`, it is still possible that end-of-file has been reached. The sender can recognize this without another DPS_GetPartialFile operation by comparing `offset + maxSize` to the size of the file retrieved by DPS_GetFileInfo.

Implementer's Note: For DPS Version 1.1, this operation will be directly mapped to a transport-dependant operation. See Appendix B.6.3.3.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getPartialFile>
   <fileID>00000001</fileID>
   <offset>00000000</offset>
   <maxSize>00000A00</maxSize>
  </getPartialFile>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getPartialFile>
   <bytesRead>00000A00</bytesRead>
  </getPartialFile>
 </output>
</dps>
```

8.2.5 DPS_GetFileList (Informative)

***Note**: This operation is not used in DPS Version 1.1, nor is it mapped to transport-specific operations. As it is a basic capability of the Storage Service, it is defined here for completeness. The operation name and parameter names are reserved for future versions.*

Parameters for DPS_GetFileList

	Parameters		Description
Input	getFileList		Parameters for DPS_GetFileList request.
		fileType	The type of file whose fileIDs are to be retrieved from the Storage Server (See section 10.5 for details). If present, this parameter restricts the output list to the set of files of this type. If absent, all fileIDs of all fileTypes are returned.
		parentFileID	The directory containing the files whose fileIDs are to be retrieved.
		maxNumIDs	The maximum number of fileIDs to be retrieved (8-character hexadecimal).
Output	result		Operation result.
	getFileList		Parameters for DPS_GetFileList response.
		fileIDs	The list of fileIDs that were retrieved.
		numIDs	Number of files that satisfy the request (8-character hexadecimal).

Table 8.2-5 Parameters for DPS_GetFileList

See section 10.14 for defined values for DPS_GetFileList.

DPS_GetFileList assumes that the Storage Server supports a hierarchical file system structure. It enables the client to ask for all files of a given fileType that are directly contained in the directory (not in any subdirectories) specified by parentFileID. The first output parameter, fileIDs, is a fileID list; the second, numIDs, is the *total* number of files that satisfy the request. If numIDs is greater than maxNumIDs, then fileIDs will only contain the first maxNumIDs files that satisfy the request.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getFileList>
   <fileType>********</fileType>
```

```
      <parentFileID>00000001</parentFileID>
      <maxNumIDs>00000003</maxNumIDs>
    </getFileList>
  </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
  <output>
    <result>********</result>
    <getFileList>
      <fileIDs>00000001 00000002 00000003</fileIDs>
      <numIDs>0000000E</numIDs>
    </getFileList>
  </output>
</dps>
```

8.2.6 DPS_GetThumb (Informative)

Parameters for DPS_GetThumb

	Parameters	Description
Input	getThumb	Parameters for DPS_ GetThumb request.
	fileID	Identifier of the file whose thumbnail image is to be read.
Output	result	Operation result.
	getThumb	Parameters for DPS_ GetThumb response.
	bytesRead	Actual number of thumbnail data bytes transferred. A return value of zero indicates there is no thumbnail information associated with the file (8-character hexadecimal).

Table 8.2-6 Parameters for DPS_GetThumb

See section 10.15 for defined values for DPS_GetThumb.

DPS_GetThumb retrieves the thumbnail image from the referenced image file.

Implementer's Note: For DPS Version 1.1, this operation will be directly mapped to a transport-dependant operation. See Appendix B.6.3.4.

Execution Examples:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getThumb>
   <fileID>00000001</fileID>
  </getThumb>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getThumb>
   <bytesRead>0000A000</bytesRead>
  </getThumb>
 </output>
```

```
</dps>
```

9. DPS Parameters

9.1 DPS Parameter Conformance

The following table indicates the conformance criteria for each operation and event (i.e., whether support for the action is Mandatory, Recommended, or Optional for each the Printer and the DSC), and for all parameters.

Note that the `result` parameter shall be placed at the beginning of the output for each DPS operation or event but it is not shown in the tables below. See section 10.3.

DPS Operation/Event	Parameter			Conformance	
				Printer	DSC
DPS_ConfigurePrintService request				M	M
	dpsVersions			M	M
	vendorName			M	M
	vendorSpecificVersion			O	O
	productName			M	M
	serialNo			R	R
DPS_ConfigurePrintService response				M	M
	dpsVersions			M	M
	printServiceAvailable			M	M
	vendorName			M	M
	vendorSpecificVersion			O	O
	productName			M	M
	serialNo			R	R
DPS_GetCapability request				M	R
	capability	qualities		M	O
		paperSizes		M	R
		paperTypes	paperSize	M	O
		fileTypes		M	R
		datePrints		M	R
		fileNamePrints		M	O
		imageOptimizes		M	O
		layouts	paperSize	M	O
		fixedSizes		M	R
		croppings		M	R
		charRepertoires **V1.1**		O	O
DPS_GetCapability response				M	R
	capability	qualities		M	O
		paperSizes		M	R
		paperTypes	paperSize	M	O
		fileTypes		M	R
		datePrints		M	R
		fileNamePrints		M	O
		imageOptimizes		M	O
		layouts	paperSize	M	O

DPS Operation/Event	Parameter		Conformance	
			Printer	DSC
		fixedSizes	M	R
		croppings	M	R
		charRepertoires **V1.1**	O	O
DPS_GetJobStatus request	no parameters		M	R
DPS_GetJobStatus response			M	R
	prtPID		R	R
	filePath		R	R
	copyID		R	R
	progress		M	R
	imagesPrinted		M	R
DPS_GetDeviceStatus request	no parameters		M	O
DPS_GetDeviceStatus response			M	O
	dpsPrintServiceStatus		M	M
	jobEndReason		R	R
	errorStatus		M	M
	errorReason		M	R
	disconnectEnable		R	R
	capabilityChanged		O	O
	newJobOK		M	M
DPS_StartJob request			M	M
	jobConfig	quality	M	O
		paperType	M	O
		paperSize	M	R
		fileType	M	R
		datePrint	M	R
		fileNamePrint	M	O
		imageOptimize	M	O
		layout	M	R
		fixedSize	M	R
		cropping	M	R
	printInfo	croppingArea	M	R
		fileID	M	M
		fileName	M	O
		date	M	R
		copies	M	R
		prtPID	R	R
		filePath	R	R
		copyID	R	R
DPS_StartJob response			M	M
DPS_AbortJob request			M	R
	abortStyle		O	O
DPS_AbortJob response			M	R
DPS_ContinueJob request			M	R
DPS_ContinueJob response			M	R

DPS Operation/Event	Parameter	Conformance	
		Printer	DSC
DPS_NotifyJobStatus request		M	R
	prtPID	R	R
	filePath	R	R
	copyID	R	R
	progress	M	R
	imagesPrinted	M	R
DPS_NotifyJobStatus response		M	R
DPS_NotifyDeviceStatus request		M	M
	dpsPrintServiceStatus	M	M
	jobEndReason	R	R
	errorStatus	M	M
	errorReason	M	R
	disconnectEnable	R	R
	capabilityChanged	O	O
	newJobOK	M	M
DPS_NotifyDeviceStatus response		M	M
DPS_GetFileID request		R	R
	basePathID	M	M
	filePath	M	M
DPS_GetFileID response		R	R
	fileID	M	M
DPS_GetFileInfo request		*	*
	fileID	*	*
DPS_GetFileInfo response		*	*
	fileType	*	*
	fileSize	*	*
	thumbFormat	*	*
	thumbSize	*	*
DPS_GetFile request		*	*
	fileID	*	*
DPS_GetFile response		*	*
	bytesRead	*	*
DPS_GetPartialFile request		*	*
	fileID	*	*
	offset	*	*
	maxSize	*	*
DPS_GetPartialFile response		*	*
	bytesRead	*	*
DPS_GetFileList request		*	*
	fileType	*	*
	parentFileID	*	*
	maxNumIDs	*	*
DPS_GetFileList response		*	*

DPS Operation/Event	Parameter	Conformance	
		Printer	DSC
	fileIDs	*	*
	numIDs	*	*
DPS_GetThumb request		*	*
	fileID	*	*
DPS_GetThumb response		*	*
	bytesRead	*	*

* These operations and parameters ***shall not*** be used in Version 1.1. Equivalent PTP functionality is used instead. See Appendix B.6.3.

Table 9.1-1 M/R/O table for parameters

10. Parameter Value Sets

Parameter value sets are the valid strings that may be used for DPS action parameters. Parameters usually appear as the content of an XML element, but they may also appear as the value of an element's attribute.

Each parameter indicates, for both the Printer and the DSC, its conformance requirements. See section 3.1 for definitions of M, R, and O.

The parameter values relevant to each parameter are described in the following subsections.

New DPS parameter values added in DPS Version 1.1 are specified in *DPS parameter values addition supplements for "CIPA DC-001-2003 Rev. 2.0 or later"*, see [Supplements]. When further addition of new parameter values occurs in DPS Version 1.1 or later, they will be specified in the supplements.

Parameter values added in the supplements are valid only if both the DSC and the Printer support DPS1.1 or later.

10.1 Parameter Format

To simplify DPS parsing, the parameters described in this section consist of 8-character hexadecimal text. There are three ways of using 8-character hexadecimal text:

Usage 1, `errorReason` parameter:

"XXXXYYZZ"

XXXX	: Major code
YY	: Minor code
ZZ	: Detail code

For YY and ZZ codes,

**	: Default
-	: Codes for Common usage
-	: Codes reserved for vendor-specific use

The DPS parser may ignore YY and ZZ codes. In this case, YY and ZZ shall be treated as though both codes are "00". The Major code doesn't have vendor-specific codes.

Usage 2, `charRepertoires` parameter:

"XXYYYYYY"

XX	: Major code
YYYYYY	: Minor code

For YYYYYY code,

******-****** : Codes for common usage

The DPS parser should not ignore the Minor code. There are no vendor specific codes for this usage.

Usage 3, other than `errorReason` and `charRepertoires` parameters:

“XXXXYYYY”

XXXX : Major code

YYYY : Minor code

For YYYY code,

**** : Default

****-**** : Codes for common usage

****-**** : Codes reserved for vendor-specific use

The DPS parser may ignore the Minor code. In such cases, YYYY shall be treated as “****”. The Major code doesn’t have vendor-specific codes.

Note: The vendor-specific codes are defined by each vendor. The dpsVendorSpecificVersion information exchanged during the DPS_ConfigurePrintService operation should be used to uniquely identify the vendor-specific Minor code values.

10.2 Default Parameter Values

If a parameter is omitted, the receiver shall behave as though the default parameter value had been sent.

10.3 Parameters for DPS Action Results

For every DPS action request, a result value is returned along with output parameters for DPS action. Parameter descriptions for the result are shown below.

Parameter			**Conformance**	
Name	**Value**	**Description**	**Printer**	**DSC**
`result`			M	M
	********	OK	M	M
	********	Not executed	M	M
	********	Operation not supported	M	M
	********	Operation not recognized	M	M

Table 10.3-1 result Parameter Values

10.3.1 result Minor Code

When the action result is "Operation not supported", the parameter descriptions for minor codes for `result` are shown below.

Parameter				**Conformance**	
Major Category	**Value**	**Detail**	**Description**	**Printer**	**DSC**
Operation not supported					
	****	none	Default	M	M
	****	unrecognized parameter	A parameter used in the request was not recognized.	R	R
	****	illegal parameter	A parameter value used in the request is not allowed.	R	R
	****	missing parameter	A required parameter is missing from the request	R	R
	****	buffer overflow	The request could not be processed because the request was bigger than the receive buffer.	R	R

Table 10.3-2 minor result Parameter Values

10.4 Parameters for DPS_ConfigurePrintService

Parameter descriptions for DPS_ConfigurePrintService are shown below.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
dpsVersions	decimal	List of DPS Versions (each individual version value has a maximum length of 8 characters including one optional decimal point.)	M	M
vendorName	string	Maximum length 64 ASCII characters	M	M
vendorSpecificVersion	decimal	Maximum length of 8 characters, including optional decimal point.	O	O
productName	string	Maximum length 64 ASCII characters.	M	M
SerialNo	string	Maximum length 64 ASCII characters.	R	R
printServiceAvailable			M	M
	********	False: Print Service is not available.	M	M
	********	True: Print Service is available.	M	M

Table 10.4-1 DPS_ConfigurePrintService Parameter Values

10.5 Parameters for DPS_GetCapability

Parameter descriptions for children of the `capability` parameter are shown below. A child element of each `capability` output parameter shall include the default value; so that DSC may select the default value to the parameters at DPS_StartJob operation (see section 10.8).

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
qualities			M	O
	********	Default quality	M	M
	********	normal	R	O
	********	draft	R	O
	********	fine	R	O
paperSizes			M	R
	********	Default record media size of Printer	M	M
	********	L (89.0 mm x 127.0 mm)	O	O
	********	2L (127.0 mm x 178.0 mm)	O	O
	********	Hagaki Postcard (100.0 mm x 148.0 mm)	O	O
	********	Card Size (54.0 mm x 85.6 mm)	O	O
	********	100 mm x 150 mm	O	O
	********	4” x 6” (101.6 mm x 152.4 mm)	O	O
	********	8” x 10” (203.2 mm x 254.0 mm)	O	O
	********	Letter (216.0 mm x 279.4 mm)	O	O
	********	11" x 17" (279.4 mm x 431.8 mm)	O	O
	********	An (n=0~9) (Note 2)	O	O
	********	A4 (210.0 mm x 297.0 mm)	O	O
	********	A6 (105 mm x 148.5 mm)	O	O
	********	Bm (m=0~9) (Note 3)	O	O
	********	89 mm Roll (L)	O	O
	********	127 mm Roll (2L)	O	O
	********	4” Roll (101.6 mm)	O	O
	********	210 mm Roll (A4)	O	O
paperTypes			M	O
(Note 1)	********	Default record media of Printer	M	M
	********	Plain Paper	O	O
	********	Photo Paper	O	O
	********	Fast Photo Paper	O	O

Table 10.5-1 Capability Parameter Values

Note 1: See section 10.5.1 for defined values for paperType Minor Code.

Note 2: Codes for A0 to A9 are assigned to ******** to ********.

Note 3: Codes for B0 to B9 are assigned to ******** to ********.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
fileTypes			M	R
	********	Default fileType of Printer	M	M
	********	Exif / JPEG Format	M	M
	********	Other Exif Format (such as Exif / TIFF)	O	O
	********	JPEG Format	R	O
	********	TIFF/EP Format	O	O
	********	FlashPix Format	O	O
	********	BMP Format	O	O
	********	CIFF Format	O	O
	********	GIF Format	O	O
	********	JFIF Format	O	O
	********	PCD Format	O	O
	********	PICT Format	O	O
	********	PNG Format	O	O
	********	TIFF Format	O	O
	********	TIFF/IT Format	O	O
	********	JP2 Format	O	O
	********	JPX Format	O	O
	********	Undefined non-image Format	O	O
	********	Association	O	O
	********	Script	O	O
	********	Executable	O	O
	********	Text	O	O
	********	HTML	O	O
	********	XHTML-Print (Note 4)	O	O
	********	DPOF	R	R
	********	AIFF	O	O
	********	WAV	O	O
	********	MP3	O	O
	********	AVI	O	O
	********	MPEG	O	O
	********	ASF	O	O
datePrints			M	R
	********	Default setting of Printer	M	M
	********	off	M	R
	********	on	R	R

Table 10.5-1 Capability Parameter Values

Note 4: See section 10.5.2 for defined values for fileTypes Minor Code.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
fileNamePrints			M	O
	********	Default setting of Printer	M	M
	********	off	M	O
	********	on	O	O
imageOptimizes			M	O
	********	Default setting of Printer	M	M
	********	off	M	O
	********	on	O	O
layouts			M	O
	********	Default Layout of Printer	M	M
	********	1-Up Bordered Printing	R	O
	********	2-Up Layout	O	O
	********	3-Up Layout	O	O
	********	4-Up Layout	O	O
	********	5-Up Layout	O	O
	********	6-Up Layout	O	O
	********	7-Up Layout	O	O
	********	8-Up Layout	O	O
	:	:		
	********	250-Up Layout	O	O
	********	Index Print	R	R
	********	1-Up Borderless(full bleed) Printing	R	R

Table 10.5-1 Capability Parameter Values

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
fixedSizes			M	R
(Note 5)	********	Default fixedSize of Printer	M	M
	********	2 ½" x 3 ¼"	O	O
	********	3.5" x 5" (127 mm x 89 mm)	O	O
	********	4" x 6"	O	O
	********	5" x 7" (178 mm x 127 mm)	O	O
	********	8" x10"	O	O
	********	254 mm x 178 mm	O	O
	********	110 mm x 74 mm	O	O
	********	89 mm x 55 mm	O	O
	********	100 mm x 148 mm	O	O
	********	6 cm x 8 cm	O	O
	********	7 cm x 10 cm	O	O
	********	9 cm x13 cm	O	O
	********	10 cm x15 cm	O	O
	********	13 cm x18 cm	O	O
	********	15 cm x 21 cm	O	O
	********	18 cm x 24 cm	O	O
	********	A4 (210 mm x 297 mm full bleed)	O	O
	********	Letter (8.5" x 11" full bleed)	O	O
croppings			M	R
	********	Default setting of Printer	M	M
	********	off	M	R
	********	on	R	R
charRepertoires **V1.1**			O	O
(Note 6)	********	Character sets of the IANA Registered Charset List (xxxxxx = hexadecimal value of MIBenum)	O	O
	********	Unicode code blocks (yyyyyy = hexadecimal value of block starting code point)	O	O

Table 10.5-1 Capability Parameter Values

Note 5: At least one parameter other than default is recommended to support for Printer. See Appendix F for reference table of common name for fixed-size

Note 6: Printers that support XHTML-Print shall support the necessary values of `charRepertoires` to represent the characters they can print. The charRepertoires Minor Code is added in DPS Version 1.1. See section 10.5.3 for defined values for charRepertoires Minor Code.

10.5.1 paperType Minor Code

DPS utilizes the IEEE ISTO Printer Working Group Standard for Media Standardized Names [PWG 5101.1] for names and descriptions of more specialized media types; these specialized media types are assigned a paperType minor code for utilization within DPS. Parameter descriptions for paperType Minor Code are shown below.

Major Category	Value	Detail	Description	Printer	DSC
paperTypes				M	R
	****	stationery	Separately cut sheets of an opaque material.	O	O
	****	stationery-coated	Separately cut sheets of an opaque material with a coating of unspecified type.	O	O
	****	stationery-inkjet	Separately cut sheets of an opaque material designed to minimize the spread of liquid inks. May be accomplished using a coating.	O	O
	****	stationery-preprinted	Separately cut sheets of an opaque material with a preprinted image.	O	O
	****	stationery-letterhead	Separately cut sheets of an opaque material with a preprinted letterhead.	O	O
	****	stationery-prepunched	Separately cut sheets of an opaque material that are punched with an unspecified hole pattern.	O	O
	****	stationery-fine	Separately cut sheets of vellum or other high quality opaque material.	O	O
	****	stationery-heavyweight	Separately cut sheets of a heavy stock opaque material.	O	O
	****	stationery-lightweight	Separately cut sheets of a light stock opaque material.	O	O
	****	transparency	Separately cut sheets of a transparent material	O	O
	****	envelope	Envelopes that can be used for conventional mailing purposes.	O	O
	****	envelope-plain	Envelopes that are not preprinted and have no windows.	O	O
	****	envelope-window	Envelopes that have windows for addressing purposes	O	O
	****	continuous	Continuously connected sheets of an opaque material – which edge is connected is not specified.	O	O
	****	continuous-long	Continuously connected sheets of an opaque material connected along the long edge.	O	O

Table 10.5-2 paperType Minor Code Parameter Values

Parameter				Conformance	
Major Category	Value	Detail	Description	Printer	DSC
paperTypes				M	R
	****	continuous-short	Continuously connected sheets of an opaque material connected along the short edge.	O	O
	****	tab-stock	Media with tabs (either pre-cut or full-cut).	O	O
	****	pre-cut- tabs	Media with tabs that are cut so that more than one tab is visible extending out beyond the edge of non-tabbed media in an Output-Document.	O	O
	****	full-cut- tabs	Media with a tab that runs the full length of the sheet so that only one tab is visible extending out beyond the edge of non-tabbed media in an Output-Document.	O	O
	****	multi-part-form	Form medium composed of multiple layers not pre-attached to one another; each sheet may be drawn separately from an input source.	O	O
	****	labels	Label stock (For example, a sheet of peel-off labels).	O	O
	****	multi-layer	Form medium composed of multiple layers which are pre-attached to one another; e.g., for use with impact printers.	O	O
	****	screen	A refreshable display.	O	O
	****	screen-paged	A refreshable display which cannot scroll.	O	O
	****	photographic	Separately cut sheets of an opaque material to produce photographic quality images. The coating is unspecified.	O	O
	****	photographic-glossy	Separately cut sheets of an opaque material that has a "glossy" coating to produce photographic quality images.	O	O
	****	photographic-high-gloss	Separately cut sheets of an opaque material that has a "high-gloss" coating to produce photographic quality images.	O	O
	****	photographic-semi-gloss	Separately cut sheets of an opaque material that has a "semi-gloss" coating to produce photographic quality images.	O	O

Table 10.5-2 paperType Minor Code Parameter Values

Parameter				Conformance	
Major Category	**Value**	**Detail**	**Description**	**Printer**	**DSC**
paperTypes				M	R
	****	photographic-satin	Separately cut sheets of an opaque material that has a “satin” coating to produce photographic quality images.	O	O
	****	photographic-matte	Separately cut sheets of an opaque material that has a “matte” coating to produce photographic quality images.	O	O
	****	photographic-film	Separately cut sheets of film used to produce photographic quality images.	O	O
	****	back-print-film	Separately cut sheet of a translucent film that the user can view with or without backlighting.	O	O
	*****	cardstock	Separately cut sheets of a heavier or stiffer opaque material than stationery.	O	O

Table 10.5-2 paperType Minor Code Parameter Values

10.5.2 fileTypes Minor Code

This is the Minor Code for XHTML-Print, the Major Code value of which is “****” as defined in Table 10.5-1 above. Each Minor Code value represents a character encoding scheme that is supported by the Printer for reading and processing XHTML-Print files. The Minor Code values used shall be the MIBenum values found in the IANA Registered Charset List, see [IANA-CS]. These Minor Code values are 4-character hexadecimal value strings. Parameter descriptions for fileTypes Minor Code are shown below.

The fileTypes Minor Code is used exclusively to indicate decoding capabilities of the Printer and the actual encoding scheme used by the DSC—even though many of the entries in the IANA Registered Charset List define a character repertoire in addition to defining an encoding scheme.

Note: The fileTypes Minor Code is added in DPS Version 1.1.

Implementer's Note:

Printer support for a character encoding scheme does not guarantee the printer’s ability to print all of the characters that can be represented in that scheme. Use the `charRepertoires` parameter to determine the Printer’s printable characters.

Parameter			Conformance	
Major Category	**Value**	**Description**	**Printer**	**DSC**
XHTML-Print			O	O
	****	Default The ”encoding” value in an XHTML-Print document is used. (Note 1)	M	M
	hexadecimal	The 4-character hexadecimal value of MIBenum in IANA Registered Charset List	O	O

	Some examples of typical character encoding scheme are shown below	
	****	UTF-8 (decimal value ****) (Note 2)
	****	Shift_JIS (decimal value ****)
	****	Extended UNIX Code Packed Format for Japanese, EUC-JP (decimal value ****)

Table 10.5-3 fileTypes Minor Code Parameter Values

Note 1: The Printer shall not set fileTypes Minor Code to only Default, but it shall specify all character encoding schemes the Printer supports. The DSC shall use the XHTML-Print file encoded by the character encoding scheme the Printer supports. And it should set the fileType parameter in DPS_StartJob to Default.

Note 2: The Printer shall support UTF-8 when XHTML-Print is supported.

10.5.3 charRepertoires Minor Code

The charRepertoires parameter represents the Printer's printable character set, that is, the actual characters that it can render on paper. DPS defines two ways for the Printer to declare its printable character repertoire.

The first, identified by Major Code "**" as defined in Table 10.5-1 above, uses the MIBenum values found in the IANA Registered Charset List, see [IANA-CS]. These Minor Code values are 6-character hexadecimal value strings. These are the same values defined for the fileTypes Minor Code, but they are used for the character repertoire that these charsets represent, *not* the character encoding scheme that they represent.

The second, identified by Major Code "**" as defined in Table 10.5-1 above, uses Unicode code blocks. The blocks defined in Unicode Version 5, and their starting code points, may be found in [UNI-CB].

Note: The charRepertoires Minor Code is added in DPS Version 1.1.

Parameter descriptions for charRepertoires Minor Code are shown below.

Parameter			**Conformance**	
Major Category	**Value**	**Description**	**Printer**	**DSC**
Character sets of the IANA Registered Charset List			O	O
	hexadecimal	The 6-character hexadecimal value of MIBenum in IANA Registered Charset List	O	O
	Some examples of the Printer's printable character sets are shown below			
	******	ISO_8859-1:1987, ISO-8859-1 (decimal value ****)		
	******	ISO_8859-2:1987, ISO-8859-2 (decimal value ****)		
	******	ISO_8859-3:1988, ISO-8859-3 (decimal value ****)		
	******	ISO_8859-4:1988, ISO-8859-4 (decimal value ****)		
	******	ISO_8859-5:1988, ISO-8859-5 (decimal value ****)		

	******	ISO_8859-6:1987, ISO-8859-6 (decimal value ****)		
	******	ISO_8859-7:1987, ISO-8859-7 (decimal value ****)		
	******	ISO_8859-8:1988, ISO-8859-8 (decimal value ****)		
	******	ISO_8859-9:1989, ISO-8859-9 (decimal value ****)		
	******	ISO-8859-10 (decimal value ****)		
	******	ISO-8859-13 (decimal value ****)		
	******	ISO-8859-14 (decimal value ****)		
	******	ISO-8859-15 (decimal value ****)		
	******	GB18030 (decimal value ****)		
	******	JIS_C6226-1983 (decimal value ****)		
	******	KS_C_5601-1987 (decimal value ****)		
	******	Big5 (decimal value ****)		
	******	TIS-620 (decimal value ****)		
Unicode code blocks			O	O
	hexadecimal	The 6-character hexadecimal value of block starting code point	O	O
	Some examples of the Printer's printable character sets are shown below			
	******	Basic Latin (identical to US-ASCII)		
	******	Latin-1 Supplement (identical to subset of IS0-8859-1 for ISO code values ≥ 0x80)		
	******	CJK Unified Ideographs Extension A		
	******	CJK Unified Ideographs		
	******	CJK Unified Ideographs Extension B		

Table 10.5-4 charRepertoires Minor Code Parameter Values

10.6 Parameters for DPS_GetJobStatus and DPS_NotifyJobStatus

Parameter descriptions for print server status for DPS_GetJobStatus and DPS_NotifyJobStatus are shown below.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
`prtPID`	decimal	A three-digit value.	R	R
`filePath`	string	An ASCII string representing a file path name as defined by [DCF].	R	R
`copyID`	decimal	A three-digit value.	R	R
`progress`	string	Of the form *N* "/" *T* where *N* and *T* are three-digit decimal integer values in the range 001 - 999.	M	R
`imagesPrinted`	decimal	A three-digit value.	M	R

Table 10.6-1 Job Status Parameter Values

10.7 Parameters for DPS_GetDeviceStatus and DPS_NotifyDeviceStatus

Parameter descriptions for Device Status for DPS_GetDeviceStatus and DPS_NotifyDeviceStatus are shown below.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
dpsPrintServiceStatus			M	M
	********	Printing: The DPS job is active and the Printer is busy printing the DPS job.	M	R
	********	Idle: The DPS job is not active.	M	R
	********	Paused: The DPS job is active and the Printer is either paused due to an error or is waiting for user intervention or input.	M	R
jobEndReason			R	R
	********	Job not ended.	M	R
	********	Job ended normally.	M	R
	********	Job ended due to DPS_AbortJob with “abortStyle” = ********.	M	R
	********	Job ended due to DPS_AbortJob with “abortStyle” = ********.	O	O
	********	Job ended due to other reason.	M	R
errorStatus			M	M
	********	No error.	M	M
	********	Warning (recoverable error).	M	M
	********	Fatal error (unrecoverable error).	M	M
errorReason			M	R
	********	No reason.	M	R
	********	Paper-related error.	R	R
	********	Ink-related error.	R	R
	********	Hardware-related error.	R	R
	********	File-related error.	R	R
disconnectEnable			R	R
	********	False: Cannot be disconnected. Printer is still accessing the DSC.	M	R
	********	True: Can be disconnected. Printer is finished accessing the DSC.	M	R
capabilityChanged			O	O
	********	False: Printer capability is unchanged.	O	O
	********	True: Printer capability has changed.	O	O
newJobOK			M	M
	********	False: Printer cannot accept a DPS_StartJob request now.	M	M
	********	True: Printer can accept a DPS_StartJob request now.	M	M

Table 10.7-1 Device Status Parameter Values

Note: Minor error reason parameters are defined in Table 10.7-2.

10.7.1 errorReason Minor Code

Parameter descriptions for errorReason Minor Code are shown below.

errorReason values: The first four characters of the errorReason parameter value indicate a major error category as defined in Table 10.7-2 above. The two characters following the major category characters provide further information about a major error. The last two characters are reserved for further details; for this version of the specification, they shall be '**'.

Parameter			Conformance	
Major Category	**Value**	**Description**	**Printer**	**DSC**
No reason			M	R
	****	No reason.	M	O
Paper-related error			R	R
	****	Paper related error (default).	R	O
	****	Paper is empty.	O	O
	****	Load Error - paper not loaded correctly.	O	O
	****	Eject Error - paper not ejected correctly.	O	O
	****	Media Error - paper not detected, paper skewed, roll paper cannot be cut or torn -off.	O	O
	****	Paper jam.	O	O
	****	Paper nearly empty.	O	O
	****	Paper type/paper size combination not supported, or paper found in Printer does not match request.	O	O
Ink-related error			R	R
	****	Ink related reason (default).	R	O
	****	Ink empty.	O	O
	****	Ink low.	O	O
	****	Waste ink error (waste-ink reservoir full).	O	O
Hardware-related error			R	R
	****	Hardware related error (default).	R	O
	****	Fatal error.	O	O
	****	Service call.	O	O
	****	Printer unavailable (performing maintenance or otherwise occupied).	O	O
	****	Printer busy (processing a job from another source, e.g., a PC).	O	O
	****	Lever error (lever position not correct).	O	O
	****	Cover open.	O	O
	****	No marking agent (e.g., missing print head).	O	O
	****	Ink Cover open.	O	O
	****	No Ink Cartridge	O	O
File-related error			R	R
	****	File related reason (default).	R	O
	****	printInfo error (specified <printInfo> is too large).	O	O
	****	File decode error (picture format of specified image cannot be decoded).	O	O

Table 10.7-2 errorReason Minor Code Parameter Values

10.7.2 disconnectEnable

The disconnectEnable parameter indicates whether or not the Printer has retrieved all of the information for the print job. If True, the DSC has the option of terminating the connection or continuing to monitor print progress.

10.7.3 capabilityChanged

The capabilityChanged parameter indicates whether or not one or more of the Printer's capabilities has changed. Upon a change in capabilities, the Printer shall set capabilityChanged to True and issue DPS_NotifyDeviceStatus. The DSC then has the option of refreshing its knowledge of Printer capabilities by issuing one or more DPS_GetCapability operation requests.

The Printer shall set capabilityChanged to False upon issuing a response to a DPS_GetCapability operation request, no matter which capability is retrieved. A DPS_NotifyDeviceStatus event shall not be triggered by the transition of capabilityChanged from True to False.

10.7.4 newJobOK

The newJobOK parameter indicates whether or not the Printer is ready to accept DPS_StartJob from the DSC. True indicates the Printer is ready to accept the next job. False indicates the Printer is not ready to accept the next job.

Printer status updates shall report the progress of the job currently printing; or, if all jobs the Print Client has sent are waiting to be printed, job progress reports shall indicate the status of the first job in the queue.

10.8 Parameters for DPS_StartJob

Parameter descriptions for DPS_StartJob are shown below. When a DPS print client sets the value of any `jobConfig` input parameter to its “Default ...“ value, the client is allowing the DPS print server to set a value for the parameter. DPS does not specify how the printer selects a value, it is decided solely by the printer vendor. It could be based on anything, including, but not limited to: 1) pre-set by the factory, 2) pre-set by a user, 3) set automatically by a sensor on the printer, 4) other settings in the printer’s current configuration.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
quality	hexadecimal	8-digit code. See section 10.5 “qualities”.	M	O
paperType	hexadecimal	8-digit code. See section 10.5 “paperTypes”.	M	O
paperSize	hexadecimal	8-digit code. See section 10.5 “paperSizes”.	M	R
fileType	hexadecimal	8-digit code. See section 10.5 “fileTypes”.	M	R
datePrint	hexadecimal	8-digit code. See section 10.5 “datePrints”.	M	R
fileNamePrint	hexadecimal	8-digit code. See section 10.5 “fileNamePrints”.	M	O
imageOptimize	hexadecimal	8-digit code. See section 10.5 “imageOptimizes”.	M	O
layout	hexadecimal	8-digit code. See section 10.5 “layouts”.	M	R
fixedSize	hexadecimal	8-digit code. See section 10.5 “fixedSizes”.	M	R
cropping	hexadecimal	8-digit code. See section 10.5 “croppings”.	M	R

Table 10.8-1 DPS_StartJob Parameter Values

Parameter			**Conformance**	
Name	**Value**	**Description**	**Printer**	**DSC**
croppingArea	hexadecimal	Specified as "“aaaa bbbb cccc dddd"”. See section 10.8.1.	M	R
fileID	hexadecimal	8-digit code. See Appendix B	M	M
fileName	string	Maximum length is 24 alphanumeric characters	M	O
date	string	Maximum length is 24 alphanumeric characters	M	R
copies	decimal	A non-negative integer containing exactly 3 digits	M	R
prtPID	decimal	A non-negative integer containing exactly 3 digits	R	R
filePath	string	See Appendix C (for DPOF restart) or Appendix H (for XHTML-Print)	R	R
copyID	decimal	A non-negative integer containing exactly 3 digits	R	R

Table 10.8-1 DPS_StartJob Parameter Values

10.8.1 Cropping

Some printers support image cropping. The area to be printed is indicated as a rectangle, with the upper left corner, width and height, specified relative to the origin (see Figure 10.8-1 below).

Cropping area values are provided as four sets of 4-character hexadecimal strings separated by spaces (E.g., `<croppingArea>0000 0000 0280 01E0</croppingArea>`. See section 7.2.5 for a complete example.).

The four hex strings define the corner, width and height of the portion of the image to be printed.

- The first string specifies the upper-left X (Horizontal) coordinate (X).
- The second string specifies the upper-left Y (Vertical) coordinate (Y).
- The third string specifies the width (W).
- The fourth string specifies the height (H).

Units of the coordinate, width and height are image pixels. In the figure, the white area will be printed and the gray will be cropped (i.e., not printed).

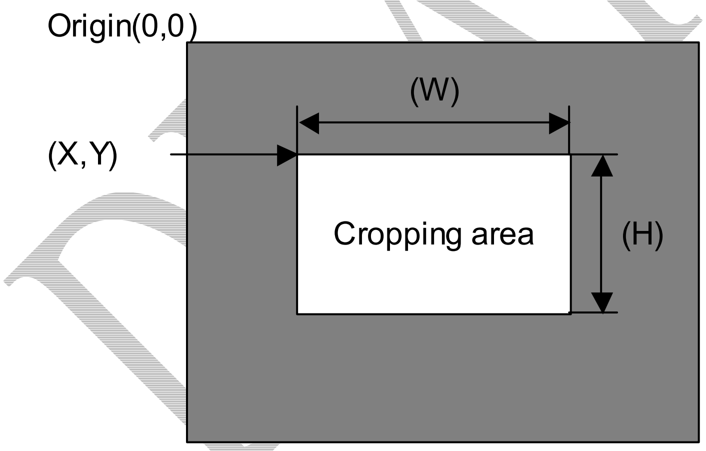

Figure 10.8-1. Cropping Area

10.8.2 Fixed Print Size

The Printer could print image(s) according to its actual print size by specifying the fixedSize parameter. In this case, the layout and fitting of images on a page are up to the Printer's capability.

For example: 4x6 inches on letter size paper can fit 3 pictures but the Printer would be required to auto rotate the image. If the Printer is incapable of rotation, it is allowed to print the images across more pages if necessary. If the paper is not large enough for the image selected then a paper error should be generated by the print server and sent to the print client.

Examples: This is acceptable output from a request to print three 4x6 pictures.

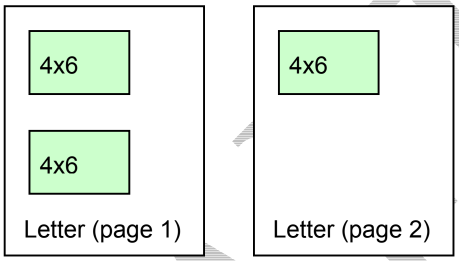

Figure 10.8-2 Fixed-size print Example

10.9 Parameters for DPS_AbortJob

Parameter descriptions for DPS_AbortJob are shown below.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
abortStyle			O	O
	********	Abort job immediately	M	R
	********	Abort job after completing the current page	O	O

Table 10.9-1 DPS_AbortJob Parameter Values

10.10 Parameters for DPS_GetFileID

Parameter descriptions for DPS_GetFileID are shown below.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
basePathID	hexadecimal	Exactly 8 characters	M	M
filePath	string	ASCII characters	M	M
fileID	hexadecimal	Exactly 8 characters	M	M

Table 10.10-1 DPS_GetFileID Parameter Values

10.11 Parameters for DPS_GetFileInfo (Informative)

Parameter descriptions for DPS_GetFileInfo are shown below.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
fileID	hexadecimal	Exactly 8 characters.	*	*
fileType	hexadecimal	8-digit code. See "fileTypes" in Table 10.5-1.	*	*
fileSize	hexadecimal	Exactly 8 characters.	*	*
thumbFormat	hexadecimal	8-digit code. See "fileTypes" in Table 10.5-1.	*	*
thumbSize	hexadecimal	Exactly 8 characters.	*	*

* DPS_GetFileInfo and these associated parameters ***shall not*** be used in DPS Version 1.1. See Appendix **B.6.3.1** for details.

Table 10.11-1 DPS_GetFileInfo Parameter Values

10.12 Parameters for DPS_GetFile (Informative)

Parameter descriptions for DPS_GetFile are shown below.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
fileID	hexadecimal	Exactly 8 characters.	*	*
bytesRead	hexadecimal	Exactly 8 characters.	*	*

* DPS_GetFile and these associated parameters ***shall not*** be used in DPS Version 1.1. See Appendix B.6.3.2 for details.

Table 10.12-1 DPS_GetFile Parameter Values

10.13 Parameters for DPS_GetPartialFile (Informative)

Parameter descriptions for DPS_GetPartialFile are shown below.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
fileID	hexadecimal	Exactly 8 characters	*	*
offset	hexadecimal	Exactly 8 characters	*	*
maxSize	hexadecimal	Exactly 8 characters	*	*
bytesRead	hexadecimal	Exactly 8 characters	*	*

* DPS_GetPartialFile and these associated parameters ***shall not*** be used in DPS Version 1.1. See Appendix **B.6.3.3** for details.

Table 10.13-1 DPS_GetPartialFile Parameter Values

10.14 Parameters for DPS_GetFileList (Informative)

Parameter descriptions for DPS_GetFileList are shown below.

Parameter			Conformance	
Name	**Value**	**Description**	**Printer**	**DSC**
fileType	hexadecimal	8-digit code. See section 10.5 “fileTypes”.	*	*
parentFileID	hexadecimal	Exactly 8 characters	*	*
maxNumIDs	hexadecimal	Exactly 8 characters	*	*
fileIDs	hexadecimal	Space-separated list of 8-character values	*	*
numIDs	hexadecimal	Exactly 8 characters	*	*

* DPS_GetFileList and these associated parameters ***shall not*** be used in DPS Version 1.1.

Table 10.14-1 DPS_GetFileList Parameter Values

10.15 Parameters for DPS_GetThumb (Informative)

Parameter descriptions for DPS_GetThumb are shown below.

Parameter			**Conformance**	
Name	**Value**	**Description**	**Printer**	**DSC**
fileID	hexadecimal	Exactly 8 characters	*	*
bytesRead	hexadecimal	Exactly 8 characters	*	*

* DPS_GetThumb and these associated parameters ***shall not*** be used in DPS Version 1.1. See Appendix B.6.3.4 for details.

Table 10.15-1 DPS_GetThumb Parameter Values

11. Event Transitions

11.1 State Transitions on Printing Events

Table 11.1-1 shows state transitions on printing events.

Previous state	New state	Event	DPS_Notify/GetJobStatus					DPS_Notify/GetDeviceStatus						
			prtPID	filePath	copyID	progress	imagesPrinted	dpsPrintServiceStatus	jobEndReason	errorStatus	errorReason	disconnectEnable	capabilityChanged	newJobOK
IN	ID	DPS_ConfigurePrintService executed.	N/A	N/A	N/A	N/A	N/A	ID	NE	UE	UE	TR	TR	FA
ID	ID	Error status changed.						-	-	UE	UE	-	-	-
	PR	DPS_StartJob executed.						PR	NE	-	-	FA	-	FA
PR	ID	Print job ended normally.	N/A	N/A	N/A	N/A	N/A	ID	EN	-	-	-	-	-
		Print job ended by DPS_AbortJob.	N/A	N/A	N/A	N/A	N/A	ID	EA	-	-	-	-	-
		Print job ended by printer.	N/A	N/A	N/A	N/A	N/A	ID	EP	-	-	-	-	-
	PR	Start of every page.	DP	DP	DP	PG	IP							
		Print job ended normally. Next job is queued.						-	EN	-	-	-	-	-
		Start of a queued print job.						-	NE	-	-	-	-	-
		Error status changed.						-	-	UE	UE	-	-	-
	PA	Printer paused.						PA	-	UE	UE	-	-	-
PA	ID	Print job ended by DPS_AbortJob.	N/A	N/A	N/A	N/A	N/A	ID	EA	-	-	-	-	-
		Print job ended by Printer.	N/A	N/A	N/A	N/A	N/A	ID	EP	-	-	-	-	-
	PR	Printer recovered from Pause.						PR	-	UE	UE	-	-	-
	PA	Error status changed.						-	-	UE	UE	-	-	-
PP	-	DPS_StartJob executed; the job is queued.						-	-	-	-	FA	-	FA
IPP	-	All print data has been retrieved.						-	-	-	-	TR	-	-
		The printer capability was updated.						-	-	-	-	-	TR	-
		The printer capability was gotten when capabilityChanged is TRUE.						-	-	-	-	-	FA	-
		Printer is able to receive new job.						-	-	-	-	-	-	TR
		Printer is unable to receive new job.						-	-	-	-	-	-	FA

IN	Initial state
ID	Idle state
PR	Print state
PA	Pause state
PP	Print / Pause
IPP	Idle / Print / Pause
-	Same as previous

DP	Up to DPOF progress	ID	Idle state
PG	Up to job progress	PR	Print state
IP	Up to images printed	PA	Pause state
N/A	Parameter value is ineffectived because job is inactive		
NE	job Not Ended	UE	Up to Error
EN	Ended Normally	TR	TRUE
EA	Ended by DPS_AbortJob	FA	FALSE
EP	Ended by Printer Reason	-	Same as previous

Table 11.1-1 State Transitions on Printing Events

In general, DPS_NotifyJobStatus and DPS_NotifyDeviceStatus serve the purpose of keeping the DSC informed about the state of the Printer (job progress, error conditions, etc.) The Printer's capabilities can also change dynamically; for example, a new media cassette may be inserted providing an additional media size and type. Since the DSC learns about the Printer's capabilities with a DPS_GetCapability request, any subsequent changes to that set of capabilities *may* set capabilityChanged to True and cause the Printer to issue a DPS_NotifyDeviceStatus, even though the capabilities that the DSC actually requested did not change. Any subsequent changes to the set of capabilities actually requested by the DSC *shall* set capabilityChanged to True and cause the Printer to issue a DPS_NotifyDeviceStatus.

After capabilityChanged was set to True, receiving next DPS_GetCapability sets capabilityChanged to False, no matter whether the capability requested by the DSC is the capability changed or not. DPS_NotifyDeviceStatus should not be issued when capabilityChanged becomes FALSE. This capabilityChanged information can be gotten by DPS_GetDeviceStatus also.

12. References

All reference citations in this document use the bracketed name as defined in column of the following table

[ABNF]	"Augmented BNF for Syntax Specifications: ABNF", D. Crocker, P. Overell, October 2005. Available at: http://www.ietf.org/rfc/rfc4234.txt
[DCF]	*Design rule for Camera File System*, Version 1.0, JEIDA-49-2-1998, December 1998. Available at: http://it.jeita.or.jp/document/publica/standard/exif/english/DCFe.pdf
[DPOF]	*DPOF (Digital Print Order Format) Specification*, Version 1.10, 17 July 2000. Available at: http://panasonic.jp/dc/dpof_110/index.html
[Exif]	*Exchangeable image file format for digital still cameras: Exif Version 2.2*, JEITA CP-3451, April 2002. Available at: http://www.jeita.or.jp/english/
[IANA-CS]	"Character Sets", IANA Registered Charset List, 28 January 2005. Available at: http://www.iana.org/assignments/character-sets
[JPEG]	"Information technology -- Digital compression and coding of continuous-tone still images: Requirements and guidelines", ISO/IEC 10918-1:1994. Available at: http://www.iso.org/
[LatinChars]	"Character Mnemonics & Character Sets", K. Simonsen, June 1992. Available at: http://www.ietf.org/rfc/rfc1345.txt
[MIMEMPX]	"The MIME Application/Vnd.pwg-multiplexed Content-Type", R. Herriot, December 2002. Available at: http://www.ietf.org/rfc/rfc3391.txt
[MIPC]	*Implementation Guidelines for Printing with Mobile Terminals*, Version 2.0, Mobile Imaging and Printing Consortium, Inc., 27 March 2006. Available at: http://www.mobileprinting.org/
[MPVCORE]	*MPV Core Specification*, Revision 1.01, 11 March 2003. Available at: http://www.osta.org/mpv/public/specs/MPVCore-Spec-1.01.pdf
[PTP]	"Photography -- Electronic still picture imaging -- Picture transfer protocol (PTP) for digital still photography devices", ISO 15740:2005. Available at: http://www.iso.org/
[PWG 5101.1]	*The Printer Working Group Standard for Media Standardized Names*, IEEE-ISTO 5101.1-2002, 26 February 2002. Available at: ftp://ftp.pwg.org/pub/pwg/standards/pwg5101.1.pdf
[SICD]	*Universal Serial Bus Still Image Capture Device Definition*, Revision 1.0, 11 July 2000. Available at: http://www.usb.org/developers/devclass_docs/usb_still_img10.pdf
[Style]	*IEEE Standards Style Manual*, April 2002. Available at: http://standards.ieee.org/guides/style/2005Style.pdf
[Supplements]	DPS parameter values addition supplements for "CIPA DC-001-2003 Rev. 2.0 or later"

[UNI-CB]	"Unicode Character Database Blocks", *Unicode 5.0*, 15 February 2006. Available at: http://www.unicode.org/Public/5.0.0/ucd/Blocks.txt *Unicode 5.0*. Available at: http://www.unicode.org/versions/Unicode5.0.0/
[URI]	"Uniform Resource Identifier (URI): Generic Syntax", T. Berners-Lee, R. Fielding, L. Masinter, January 2005. Available at: http://www.ietf.org/rfc/rfc3986.txt
[USB]	*Universal Serial Bus Specification*, Revision 2.0, 27 April 2000. Available at: http://www.usb.org/developers/docs/usb_20_05122006.zip
[XHTML-Print]	*XHTML-Print*, Melinda Grant, Jim Bigelow, eds., W3C Recommendation. Available at: http://www.w3.org/TR/xhtml-print/
[XML]	*Extensible Markup Language (XML) 1.0 (Fourth Edition)*, Tim Bray, Jean Paoli, C. M. Sperberg-McQueen, Eve Maler, François Yergeau, eds., W3C Recommendation, 4 February 2004. Available at: http://www.w3.org/TR/2006/REC-xml-20060816
[XML Schema]	*XML Schema Part 0: Primer, Second Edition*, David C. Fallside, Priscilla Walmsley, eds. W3C Recommendation, 28 October 2004. Available at: http://www.w3.org/TR/2004/REC-xmlschema-0-20041028 *XML Schema Part 1: Structures, Second Edition*, Henry S. Thompson, David Beech, Murray Maloney, Noah Mendelsohn, W3C Recommendation, 28 October 2004. Available at: http://www.w3.org/TR/2004/REC-xmlschema-1-20041028 *XML Schema Part 2: Data Types, Second Edition*, Paul V. Biron, Ashok Malhotra, W3C Recommendation, 28 October 2004. Available at: http://www.w3.org/TR/2004/REC-xmlschema-2-20041028

Appendix A. XML Schema (Normative)

XML schema is normative. It is required to visit CIPA’s web-site to refer the latest XML schema for DPS Version 1.1 at http://www.cipa.jp/dps/schema/.

Appendix B. DPS Usage of USB and PTP (Normative)

B.1. Overview

This section describes how DSCs and Printers shall implement DPS over the USB physical layer with the Still Image Capture Device Class PTP protocol.

B.2. DSC PTP Implementation Requirements

B.2.1. PTP Reference Specifications

When a DPS system operates under PTP, it shall comply with [PTP].

"Photography-Electronic still picture imaging-Picture transfer protocol (PTP) for digital still photography devices" of ISO15740:2005 contains details of Operation, Response, Event, and ObjectFormat codes required by the initiator (Printer) and the responder (DSC). Also of special note is [SICD]. The following list shows the necessary Operation, Response, and Event codes for the system.

B.2.2. PTP Operation Codes

Operation Code	Operation Name	DPS	PTP
0x1001	GetDeviceInfo	Mandatory	Mandatory
0x1002	OpenSession	Mandatory	Mandatory
0x1003	CloseSession	Mandatory	Mandatory
0x1004	GetStorageIDs	Mandatory	Mandatory
0x1005	GetStorageInfo	Mandatory	Mandatory
0x1006	GetNumObjects	Mandatory	Pull
0x1007	GetObjectHandles	Mandatory	Pull
0x1008	GetObjectInfo	Mandatory	Pull
0x1009	GetObject	Mandatory	Pull
0x100A	GetThumb	Mandatory	Pull
0x100C	SendObjectInfo	Mandatory	Push
0x100D	SendObject	Mandatory	Push
0x101B	GetPartialObject	Mandatory	Optional

Table B.2-1 Operation Codes

DPS requires those PTP operations shown as “Mandatory” for DPS in the table above. (Adopted from ISO 15740:2005, section 14, “Conformance-Section.”)

The GetNumObjects and GetObjectHandles have two optional parameters in Standard PTP. These two parameters are mandatory for DPS-enabled DSCs and shall be fully implemented.

B.2.3. PTP Response Codes

Response Code	Description
0x2001	OK
0x2003	Session Not Open
0x2004	Invalid TransactionID
0x2005	Operation Not Supported
0x2006	Parameter Not Supported
0x2007	Incomplete Transfer
0x2008	Invalid StorageID
0x2009	Invalid ObjectHandle
0x200B	Invalid Object Format Code
0x200C	Store Full
0x200E	Store Read-Only
0x200F	Access Denied
0x2010	No Thumbnail Present
0x2013	Store Not Available
0x2014	Specification By Format Unsupported
0x2015	No Valid ObjectInfo
0x2016	Invalid Code Format
0x2019	Device Busy
0x201A	Invalid ParentObject
0x201D	Invalid Parameter
0x201E	Session Already Open

Table B.2-2 Response Codes

B.2.4. PTP Event Codes

Event Code	Event Name	Conformance
0x4002	ObjectAdded	O
0x4003	ObjectRemoved	O
0x4004	StoreAdded	O
0x4005	StoreRemoved	O
0x4007	ObjectInfoChanged	O
0x4009	RequestObjectTransfer	M

Table B.2-3 Event Codes

RequestObjectTransfer is the only mandatory event.

B.3. DPS Discovery

Over PTP/USB, the two devices discover each other in the following manner:

Upon a USB connect event, the DSC will prepare a virtual file named "DDISCVRY.DPS". The contents of this file are undefined; it is recommended that it is empty. The ObjectInfo Dataset shall contain the field value described below:

ObjectInfo Dataset	
Field Name	**Value**
ObjectFormat	0x3002
Filename	“DDISCVRY.DPS”

- The Printer sends GetNumObjects and GetObjectHandles with the Destination Storage Parameter (Operation Parameter 1) for all storage(0x********) and the ObjectFormatCode (Operation Parameter 2) for script(0x3002).
- The Printer sends GetObjectInfo with the ObjectHandle (Operation Parameter 1) that was got by the GetObjectHandles.
- The Printer will retrieve file info, looking for the file whose name is DDISCVRY.DPS. If found, the Printer has discovered a DPS device.
- The Printer next prepares a virtual file named "HDISCVRY.DPS". The contents of this file are undefined; it is recommended that it is empty. The ObjectInfo Dataset shall contain the field value described below:

ObjectInfo Dataset	
Field Name	**Value**
ObjectFormat	0x3002
Filename	“HDISCVRY.DPS”

- The Printer sends SendObjectInfo with the Destination Storage ID parameter (Operation Parameter 1) and the Parent ObjectHandle parameter (Operation Parameter 2) of the SendObjectInfo shall set to zero.
- The device interprets the receipt of this ObjectInfo as discovery of a DPS device.

The sequence is depicted in detail in the following graphic:

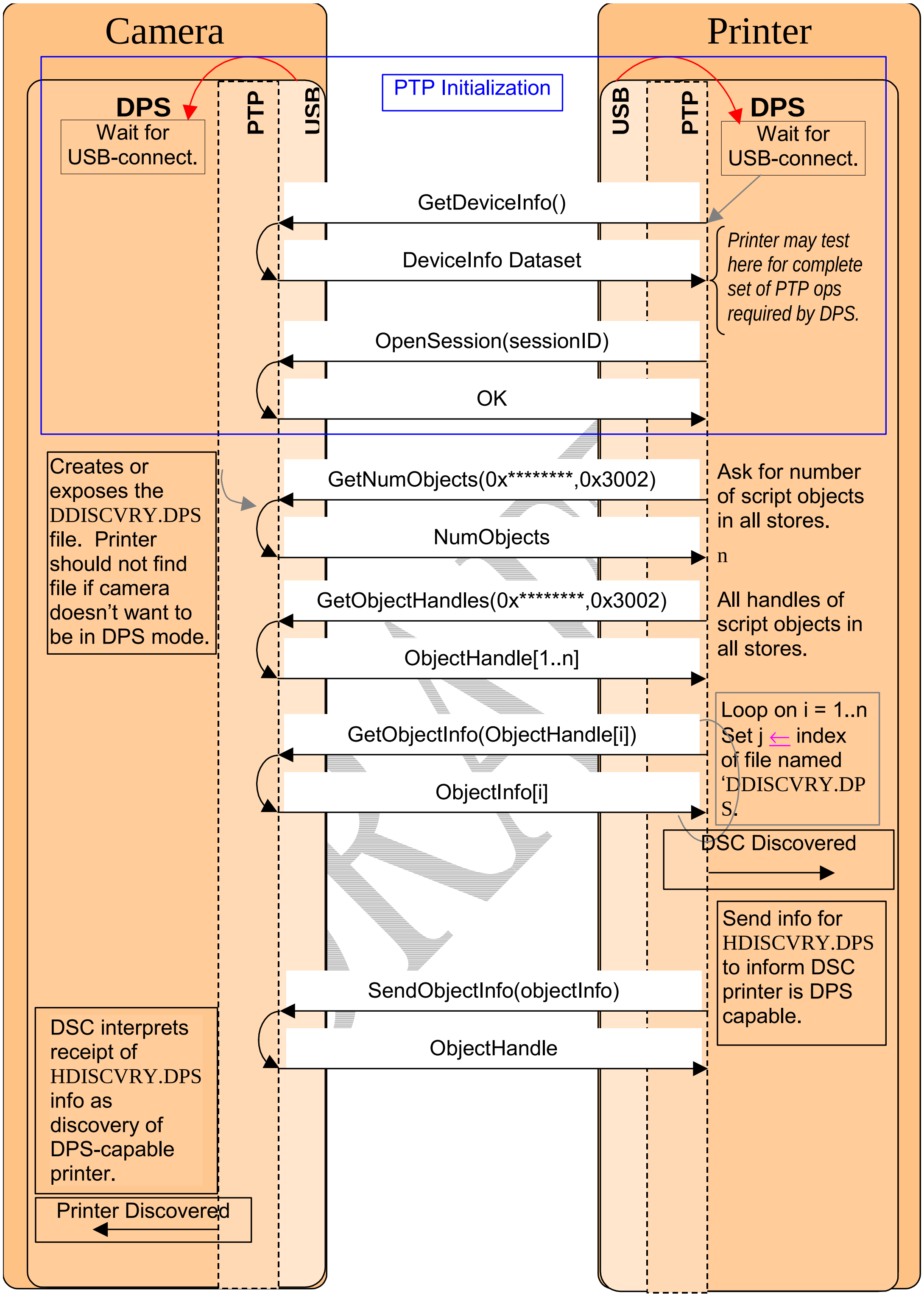
Camera
Printer
PTP Initialization
DPS
Wait for USB-connect.
PTP
USB
USB
PTP
DPS
Wait for USB-connect.
GetDeviceInfo()
DeviceInfo Dataset
Printer may test here for complete set of PTP ops required by DPS.
OpenSession(sessionID)
OK
Creates or exposes the DDISCVRY.DPS file. Printer should not find file if camera doesn't want to be in DPS mode.
GetNumObjects(0x********,0x3002)
Ask for number of script objects in all stores.
NumObjects
n
GetObjectHandles(0x********,0x3002)
All handles of script objects in all stores.
ObjectHandle[1..n]
Loop on i = 1..n
Set j ← index of file named 'DDISCVRY.DPS.
GetObjectInfo(ObjectHandle[i])
ObjectInfo[i]
DSC Discovered
Send info for HDISCVRY.DPS to inform DSC printer is DPS capable.
SendObjectInfo(objectInfo)
ObjectHandle
DSC interprets receipt of HDISCVRY.DPS info as discovery of DPS-capable printer.
Printer Discovered

Figure B.3-1 DPS Discovery

B.4. Special DSC PTP Implementations for DPS

B.4.1. SendObjectInfo/SendObject

When a DPS-compliant DSC receives a SendObjectInfo command and an ObjectInfo dataset that has the fields listed below set to the values shown, it shall assume that the SendObject data to follow is either a DPS request or a DPS response. It shall then pass the SendObject data to the DPS layer in the DSC.

ObjectInfo Dataset	
Field Name	**Value**
ObjectFormat	0x3002
Filename	"HREQUEST.DPS"

ObjectInfo Dataset	
Field Name	**Value**
ObjectFormat	0x3002
Filename	"HRSPONSE.DPS"

The Destination Storage ID parameter (operation parameter 1) and the Parent ObjectHandle parameter (operation parameter 2) of the SendObjectInfo command shall be set to zero.

Any SendObjectInfo/SendObject command that does not meet the above criteria should be handled per the PTP specification.

The DSC shall not leave these files (especially any of the discovery files) on removable media that could then get inserted into a non-DPS DSC.

B.4.2. GetNumObjects/GetObjectHandles/GetObjectInfo/GetObject

When the DSC wants to send a DPS request or a DPS response to the Printer, it should create an object with an ObjectInfo dataset that includes the following required information:

ObjectInfo Dataset	
Field Name	**Value**
ObjectFormat	0x3002
Filename	"DREQUEST.DPS"

ObjectInfo Dataset	
Field Name	**Value**
ObjectFormat	0x3002
Filename	"DRSPONSE.DPS"

The other object dataset fields shall be filled in with appropriate values.

The DSC should then send a RequestObjectTransfer event with the ObjectHandle of this new object to the Printer which will trigger the Printer to perform a GetObjectInfo/GetObject sequence.

When the DSC receives the GetObject, it should send the appropriate XML request (or response) to the Printer.

B.5. Special Printer PTP Implementation Considerations for DPS

B.5.1. Mapping of DPS fileIDs to PTP ObjectHandles

The DPS fileID is a 32-bit quantity that uniquely identifies any DPS object; i.e., there is a one-to-one correspondence between DPS objects and DPS fileIDs. No other meaning is assigned to the fileID. The PTP ObjectHandle is used as the DPS fileID.

B.5.2. Sending XML Request/Response Data to a DSC

DPS request/response XML packets will be sent to the DSC via the PTP SendObjectInfo/SendObject sequence.

The ObjectInfo DataSet used by SendObjectInfo for these operations shall include the following fields:

ObjectInfo Dataset	
Field Name	**Value**
ObjectFormat	0x3002
Filename	“HREQUEST.DPS”

ObjectInfo Dataset	
Field Name	**Value**
ObjectFormat	0x3002
Filename	“HRSPONSE.DPS”

The Destination StorageID parameter (Operation Parameter1) and the Parent ObjectHandle parameter (Operation Parameter2) of the SendObjectInfo shall be set to zero.

B.5.3. Event Handling

The Printer should respond to a RequestObjectTransfer event by performing GetObjectInfo/GetObject on the requested object. If the object is one of the special files (D*xxxxxxx*.DPS), it should pass this info on to it’s DPS system.

The Printer shall monitor all PTP events coming from the DSC and respond in a timely manner.

B.6. Implementation of DPS Actions Over PTP

B.6.1. Overview

Most DPS actions will be passed over PTP/USB in their original XML format using PTP merely as the conduit to pass the XML script as a file object. Some DPS actions either because they pass large amounts of binary data or because they would be more efficient to implement using the standard PTP operations are not passed through to the DPS layer on the DSC but rather implemented entirely on the Printer side using lower level PTP operations (these actions will be referred to as “special optimized” actions below).

The following table specifies which actions are "special optimized" actions. All other actions are "pass-through" actions.

Action Name	Description
DPS_GetFileInfo	Get file information
DPS_GetFile	Get file from Storage
DPS_GetPartialFile	Get partial file from Storage
DPS_GetFileList	Get file list from Storage
DPS_GetThumb	Get Thumbnail data

Table B.6-1 DPS special optimized actions

B.6.2. DPS Pass-Through Actions

B.6.2.1. Printer-to-DSC DPS Actions

PTP Sequence:

DSC		Printer
DPS Request (Printer-to-DSC): “HREQUEST.DPS”		
←	SendObjectInfo (ObjectInfoDataset) request, data	←
→	SendObjectInfo (ObjectHandle) response	→
←	SendObject (XML request data) request, data	←
→	SendObject response	→
→	ObjectRemoved (ObjectHandle) event (Note)	→
DPS Response (DSC-to-Printer): “DRSPONSE.DPS”		
→	ObjectAdded (ObjectHandle) event (Note)	→
→	RequestObjectTransfer (ObjectHandle) event	→
←	GetObjectInfo (ObjectHandle) request	←
→	GetObjectInfo data, response	→
←	GetObject (ObjectHandle) request	←
→	GetObject (XML response data) data, response	→
→	ObjectRemoved (ObjectHandle) event (Note)	→

Note: Recommended

Table B.6-2 Printer-to-DSC actions

Printer Implementation Notes:

- The DPS Printer must be watching for asynchronous events so that it can catch and act upon the RequestObjectTransfer event.
- The DPS Printer should format the request data in the method described in Appendix B.5.2.

DSC Implementation Notes:

- The DSC should format the DPS response XML as a virtual file. This virtual file should be handled appropriately as discussed in Appendix B.4.
- The DSC will recognize the SendObjectInfo information as a DPS request/response and pass the data on to the DPS controller in the DSC as described in Appendix B.4.

B.6.2.2. DSC-to-Printer DPS Actions

PTP Sequence:

DSC		Printer
DPS Request (DSC-to-Printer): “DREQUEST.DPS”		
→	ObjectAdded (ObjectHandle) event (Note)	→
→	RequestObjectTransfer (ObjectInfoHandle) event	→
←	GetObjectInfo (ObjectInfoHandle) request	←
→	GetObjectInfo data, response	→
←	GetObject (ObjectHandle) request	←
→	GetObject (XML request data) data, response	→
→	ObjectRemoved (ObjectHandle) event (Note)	→
DPS Response (Printer-to-DSC): “HRSPONSE.DPS”		
←	SendObjectInfo (ObjectInfoDataset) request, data	←
→	SendObjectInfo (ObjectHandle) response	→
←	SendObject (XML response data) request, data	←
→	SendObject response	→
→	ObjectRemoved (ObjectHandle) event (Note)	→

Note: Recommended

Table B.6-3 DSC-to-Printer actions

Printer Implementation Notes:

- The DPS Printer must be watching for asynchronous events so that it can catch and act upon the RequestObjectTransfer event.
- The DPS Printer should format the response data in the method described in Appendix B.5.2.

DSC Implementation Notes:

- The DSC should format the DPS request XML as a virtual file. This virtual file should be handled appropriately as discussed in Appendix B.4.
- The DSC will recognize the SendObjectInfo information as a DPS request/response and pass the data on to the DPS controller in the DSC as described in Appendix B.4.

B.6.2.3. Error Handling

Normally the response XML data is just the response data returned from the DPS action. If a DPS "pass-through" action encounters a PTP error during the sequence of the above actions, the DPS system should format the appropriate error response indicating that an error occurred.

Execution example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
  <output>
   <result>********</result> <!—Operation not executed -->
  </output>
 </dps>
```

B.6.3. DPS Special Optimized Actions

The operations discussed in this section are virtual operations for DPS Version 1.1; that is, the DPS operation request is not sent from the Printer to the DSC. Instead, the Printer issues a sequence of PTP operations that accomplish the equivalent result.

B.6.3.1. DPS_GetFileInfo

DPS operation:

```
(fileType, fileSize, thumbFormat, thumbSize) ← DPS_GetFileInfo
(fileID)
```

PTP sequence:

```
ptpObjectHandle ← fileID
ptpObjectInfoDataset ← GetObjectInfo (ptpObjectHandle)

fileType ← ObjectFormat field of ptpObjectInfoDataset
fileSize ← ObjectCompressedSize field of ptpObjectInfoDataset
thumbFormat ← ThumbFormat field of ptpObjectInfoDataset
thumbSize ← ThumbCompressedSize field of ptpObjectInfoDataset
```

B.6.3.2. DPS_GetFile

DPS operation:

```
(bytesRead) ← DPS_GetFile(fileID)
```

PTP sequence:

```
ptpObjectHandle ← fileID
```

```
GetObject(ptpObjectHandle)
bytesRead ← actual size of object that was transferred
```

B.6.3.3. DPS_GetPartialFile

DPS operation:

```
(bytesRead) ← DPS_GetPartialFile(fileID, offset, maxSize)
```

PTP sequence:

```
ptpObjectHandle ← fileID
GetPartialObject(ptpObjectHandle, offset, maxSize)
    The position of the file to be read is specified by offset.
    The maximum number of bytes requested is specified by maxSize.

bytesRead ← actual number of bytes transferred
```

B.6.3.4. DPS_GetThumb

DPS operation:

```
(bytesRead) ← DPS_GetThumb(fileID)
```

PTP sequence:

```
ptpObjectHandle ← fileID
GetThumb (ptpObjectHandle)
bytesRead ← actual size of object that was transferred
```

B.7. Collision solution

A request and a response of DPS pass-through actions are implemented as different PTP operations. As discussed in section 6.5, this may cause collisions.

Meanwhile a request and a response of DPS Special optimized actions are implemented within one PTP operation, which does not cause collisions.

For DPS pass-through actions, both the DSC and the Printer may receive unexpected requests. But if both sides would throw away each unexpected request, it increases the probability of another collision. When one side throws away the unexpected request and waits for the expected response from the other side, the collision is cancelled and the other side should terminate the outstanding request.

For the easy implementation, DSC is suitable to throw away unexpected response and Printer shall terminate the last request. In Figure B.7-1, DSC sends DPS Request B and waits for Response B. Printer sends DPS Request A before getting DPS Request B. Then DSC shall ignore (discard) unexpected DPS Request A and wait for DPS Response B.

Meanwhile Printer gets DPS Request B before getting expected DPS Response A. Printer shall terminate DPS Request A and process DPS Request B. After sending DPS Response B, Printer should send DPS Request A again if necessary.

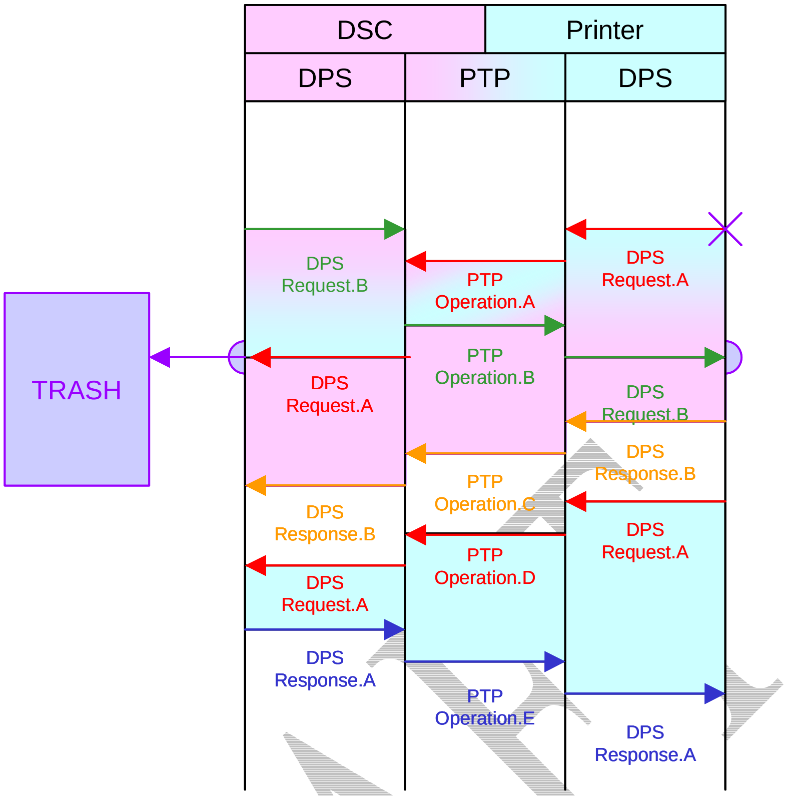

Figure B.7-1 Collision solution

To avoid other collision scenarios, when utilizing PTP as the transport layer (as is the case for version 1.1), the Printer shall never have two outstanding requests. The response to the first request shall be received before the second request may be sent. For example, the Printer might issue a DPS_GetFileID request, and then find it needs to issue a DPS_NotifyDeviceStatus request. The Printer shall postpone issuing the DPS_NotifyDeviceStatus request until the response to the DPS_GetFileID request has been received.

Appendix C. DPS Support for DPOF 1.1 (Normative)

Digital photograph print services that utilize Digital Print Order Format (DPOF, see [DPOF]) have been widely used in the DSC world. The DPS system supports the DPOF print capability and defines two ways of processing DPOF Auto Print order specified by a DPOF file, AUTPRINT.MRK.

1) In case a DSC has a DPOF parser:

The DSC analyzes the DPOF file and generates a regular DPS script. The script is transferred to the Printer and processed in it. In this case, the Printer can retrieve each image file by specifying the DPS fileID described in the operation request.

2) In case a Printer has a DPOF parser:

The DSC starts a DPOF Auto Print job by sending a DPS script that contains the DPS fileID of the DPOF file. The Printer retrieves the object of the DPOF file by specifying the DPS fileID from the DSC. The Printer then parses the DPOF file and processes it, retrieving the necessary image files. In this case, each image file is described in the DPOF file by a file path, not a DPS fileID; so the Printer cannot get the object of each image file using its file path because DPS adopts a handle-based system to specify a particular file object. Consequently, this specification defines a way of converting from a particular file path to its DPS fileID.

DPOF support is only provided for DCF-compliant DPOF implementations. In particular, implementations that utilize non-compliant characters may not function as expected.

This section describes the procedure of case 2) on how to process a DPOF Auto Print job in the Printer.

C.1. Implementation of DPOF Auto Print Using DPS

C.1.1. DPS Operations Used

The following operations are required to process DPOF Auto Print jobs. The DPS system defines an operation, DPS_GetFileID, to convert a file path of an image file as provided in the DPOF file, to a DPS fileID, on the occasion when using a DPOF parser in the Printer. See section 8.2.1 for more information about DPS_GetFileID.

- DPS_ConfigurePrintService
- DPS_GetCapability
- DPS_StartJob
- DPS_GetPartialFile/DPS_GetFile
- DPS_GetFileID
- DPS_GetDeviceStatus/DPS_NotifyDeviceStatus
- DPS_GetJobStatus/DPS_NotifyJobStatus

C.1.2. Processing sequence for DPOF Auto Print

C.1.2.1. DPOF capability confirmation

Before starting DPOF processing, the DSC should determine whether or not the Printer has a DPOF parser and supports DPOF Auto Print by getting the list of `fileType` parameters from the Printer. If the Printer supports DPOF Auto Print, it returns `fileType` parameters which include "********" (DPOF).

An execution example is shown below:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getCapability>
   <capability>
    <fileTypes/>
   </capability>
  </getCapability>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getCapability>
   <capability>
    <fileTypes>******** ********</fileTypes>
   </capability>
  </getCapability>
 </output>
</dps>
```

C.1.2.2. Primary process

This section describes how a DPOF Auto Print job is processed in the DPS system. The following is an example showing a typical file system on a DSC that complies with DCF and a typical DPOF file. Note that sample DPS scripts in this explanation are assuming the following file structure and DPOF file.

```
/MISC/
/MISC/AUTPRINT.MRK
/DCIM/
/DCIM/100ABCDE/
/DCIM/100ABCDE/FGHI0001.JPG
/DCIM/100ABCDE/FGHI0002.JPG
```

Figure C.1-1 DCF File System Directory Example

DPS fileID	File or Folder Description
00000001	/MISC folder
00000002	/MISC/AUTPRINT.MRK
00000003	/DCIM folder
00000004	/DCIM/100ABCDE folder
00000005	/DCIM/100ABCDE/FGHI0001.JPG
00000006	/DCIM/100ABCDE/FGHI0002.JPG

Table C.1-1 DCF File System Example

```
[HDR]
GEN REV = 01.00
GEN CRT = "Ultra Shot"
GEN DTM = 2002:09:06:19:08:40

[JOB]
PRT PID = 001
PRT TYP = STD
PRT QTY = 003
IMG FMT = EXIF2 -J
<IMG SRC = "../DCIM/100ABCDE/FGHI0001.JPG">

[JOB]
PRT PID = 002
PRT TYP = STD
PRT QTY = 003
IMG FMT = EXIF2 -J
<IMG SRC = "../DCIM/100ABCDE/FGHI0002.JPG">
```

Figure C.1-2 AUTPRINT.MRK Example

1) Start a DPOF Auto Print Job

The DSC starts a DPOF Auto Print job by issuing the DPS_StartJob operation and passes the DPS fileID for the DPOF file, with the `fileID` parameter to the Printer.

An execution example is shown below:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
 <startJob>
   <jobConfig>
    <fileType>********</fileType> <!-- fileType:  ******** (DPOF) -->
   </jobConfig>
   <printInfo>
    <fileID>00000002</fileID>
   </printInfo>
  </startJob>
 </input>
</dps>
```

2) Get the DPOF file

The Printer issues DPS_GetFile or DPS_GetPartialFile with the DPS fileID of the DPOF file, 00000002, to obtain its object from the DSC's Storage Server.

3) Get a DPS fileID for an image file

Image files are referenced in a DPOF file by a relative path name from a known directory in the DSC. Since DPS is file system independent, it requires a special operation to ask the DSC to convert a file path of an image file specified in the DPOF file into a valid DPS fileID. The operation is DPS_GetFileID. The Printer calls it to get a usable DPS fileID from the DPOF pathname.

An execution example is shown below:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getFileID>
   <basePathID>00000002</basePathID>
   <filePath>../DCIM/100ABCDE/FGHI0001.JPG</filePath>
  </getFileID>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getFileID>
   <fileID>00000005</fileID>
  </getFileID>
 </output>
</dps>
```

The number of 00000005 that is framed in by the `fileID` parameter is the DPS fileID of the file path requested from the Printer.

4) Get an image file to be printed

The Printer then retrieves the image file for printing using DPS_GetFile or DPS_GetPartialFile for the DPS fileID returned by DPS_GetFileID.

5) Repeat 3)-4) if multiple copies and/or multiple image files to be printed are specified in the DPOF file.

Figure C.1-3 shows the state transition and sequence diagram for processing a DPOF Auto Print job.

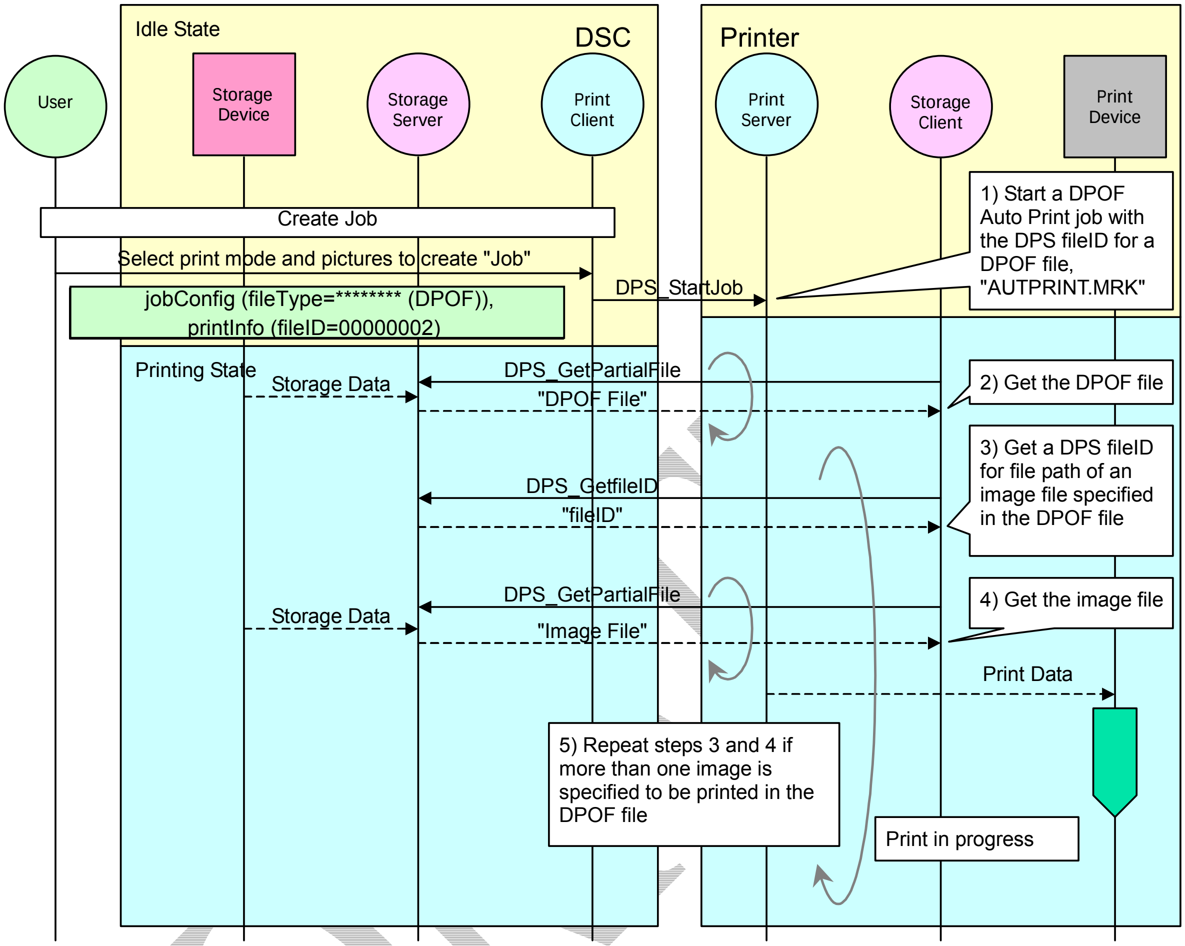
Idle State
DSC
Printer
User
Storage Device
Storage Server
Print Client
Print Server
Storage Client
Print Device
Create Job
Select print mode and pictures to create "Job"
jobConfig (fileType=******** (DPOF)), printInfo (fileID=00000002)
DPS_StartJob
1) Start a DPOF Auto Print job with the DPS fileID for a DPOF file, "AUTPRINT.MRK"
Printing State
Storage Data
DPS_GetPartialFile
"DPOF File"
2) Get the DPOF file
DPS_GetfileID
"fileID"
3) Get a DPS fileID for file path of an image file specified in the DPOF file
Storage Data
DPS_GetPartialFile
"Image File"
4) Get the image file
Print Data
5) Repeat steps 3 and 4 if more than one image is specified to be printed in the DPOF file
Print in progress

Figure C.1-3 Print Transitions on DPOF Processing

C.1.3. DPOF Auto Print Status

When the DPOF file is parsed by the Printer, not by the DSC, the DSC may wish to track the print progress in detail. With the DPS_NotifyJobStatus event, the Printer sends the current job progress to the DSC as each page begins to print, so that the DSC can monitor print progress and also provide the capability to re-start an interrupted DPOF Auto Print job. The DSC also can get the restart information by using DPS_GetJobStatus.

Execution example:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
```

```
 <input>
 <notifyJobStatus>
  <prtPID>001</prtPID>
  <filePath>../DCIM/100ABCDE/FGHI0001.JPG</filePath>
  <copyID>002</copyID>
  <progress>002/006</progress>
  </notifyJobStatus>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <notifyJobStatus/>
 </output>
</dps>
```

C.1.4. DPOF Auto Print Restart

To enable the restart mechanism, the Printer shall guarantee that image files specified in a DPOF file are printed in the same order every time if the provided DPOF file is identical. This behavior is Optional for a Printer that supports DPOF processing.

In order to utilize the restart mechanism, the DSC needs to retain the job status information provided by the Printer at the start of every page in the DPS_NotifyJobStatus event until an outstanding job has completed successfully or aborted. This capability is Recommended for a DSC that supports DPOF processing. Additionally, the DSC needs to determine whether or not the Printer supports the DPOF restart feature. If the Printer supports the DPOF restart feature, it shall include the `prtPID`, `filePath`, and `copyID` parameters for the DPS_GetJobStatus and DPS_NotifyJobStatus operations. If the Printer does not include the `prtPID`, `filePath`, and `copyID` parameters in these operations, the DSC shall determine that the Printer does not support DPOF restart.

If a DPOF Auto Print job is aborted (due to a paper jam for instance), re-starting the DPOF Auto Print job from the beginning after clearing the error will waste paper and ink. Utilizing print progress status sent with the DPS_NotifyJobStatus event, the DSC can restart the job at or near the point it was terminated. The DSC sends a DPS_StartJob operation with `prtPID`, `filePath`, and `copyID`, all of which were remembered from the last DPS_NotifyJobStatus event sent by the Print Server before the abort occurred. Then the DPOF Auto Print Job will be restarted such that the first page printed is the last page that began to print before the initial job was aborted. An execution example is shown below:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <startJob>
   <jobConfig>
    <fileType>********</fileType>
   </jobConfig>
   <printInfo>
    <fileID>00000002</fileID>
    <prtPID>001</prtPID>
    <filePath>../DCIM/100ABCDE/FGHI0001.JPG</filePath>
    <copyID>002</copyID>
   </printInfo>
  </startJob>
 </input>
</dps>
```

Figure C.1-4 shows the state transition and sequence diagram for re-starting a DPOF Auto Print job.

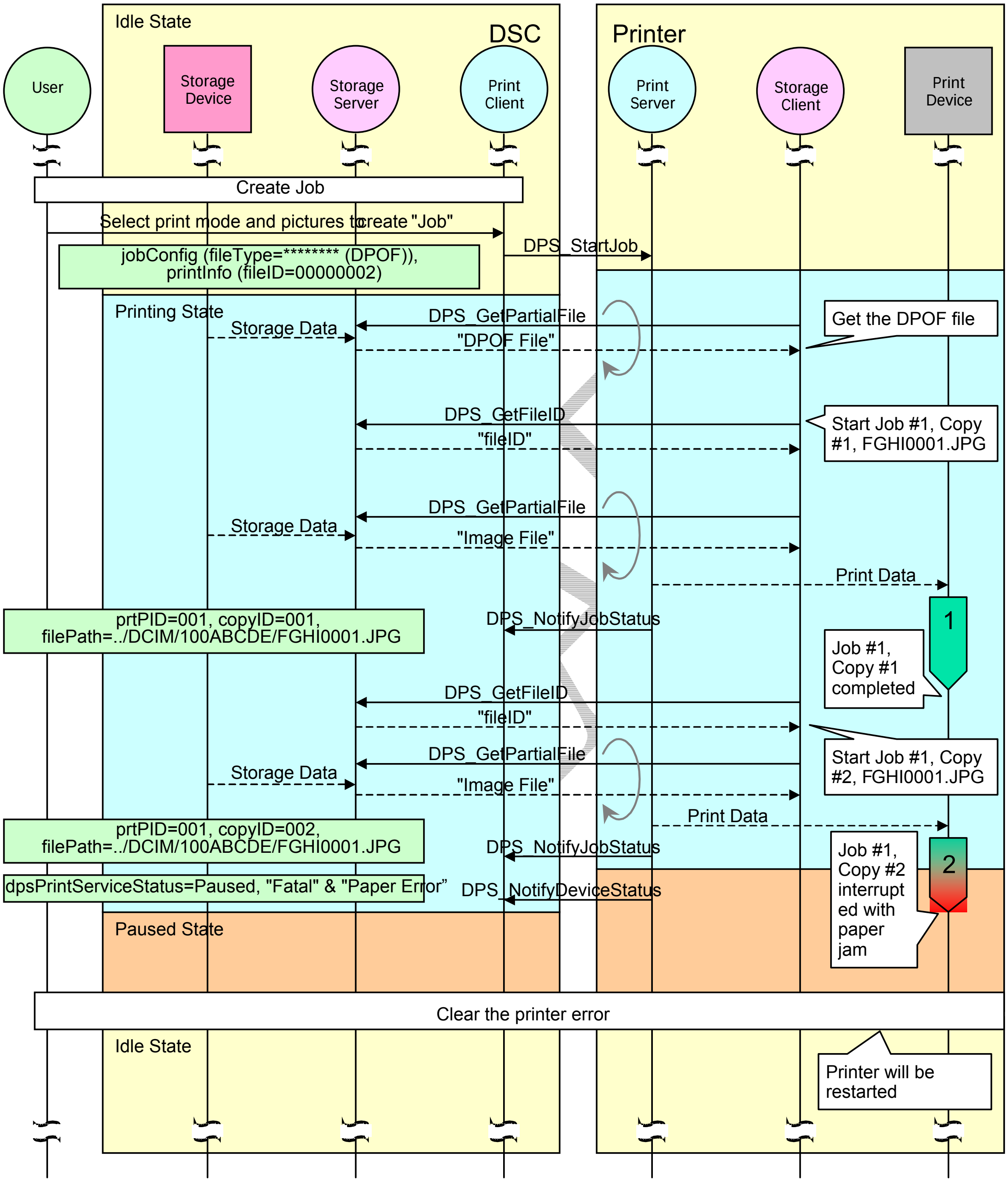
Idle State
DSC
Printer
User
Storage Device
Storage Server
Print Client
Print Server
Storage Client
Print Device
Create Job
Select print mode and pictures to create "Job"
DPS_StartJob
jobConfig (fileType=******** (DPOF)), printInfo (fileID=00000002)
Printing State
Storage Data
DPS_GetPartialFile
"DPOF File"
Get the DPOF file
DPS_GetFileID
"fileID"
Start Job #1, Copy #1, FGHI0001.JPG
DPS_GetPartialFile
Storage Data
"Image File"
Print Data
1
prtPID=001, copyID=001, filePath=../DCIM/100ABCDE/FGHI0001.JPG
DPS_NotifyJobStatus
Job #1, Copy #1 completed
DPS_GetFileID
"fileID"
Start Job #1, Copy #2, FGHI0001.JPG
DPS_GetPartialFile
Storage Data
"Image File"
Print Data
2
prtPID=001, copyID=002, filePath=../DCIM/100ABCDE/FGHI0001.JPG
DPS_NotifyJobStatus
dpsPrintServiceStatus=Paused, "Fatal" & "Paper Error"
DPS_NotifyDeviceStatus
Job #1, Copy #2 interrupt ed with paper jam
Paused State
Clear the printer error
Idle State
Printer will be restarted

Figure C.1-4 Restarting a DPOF AutoPrint Job (Part 1)

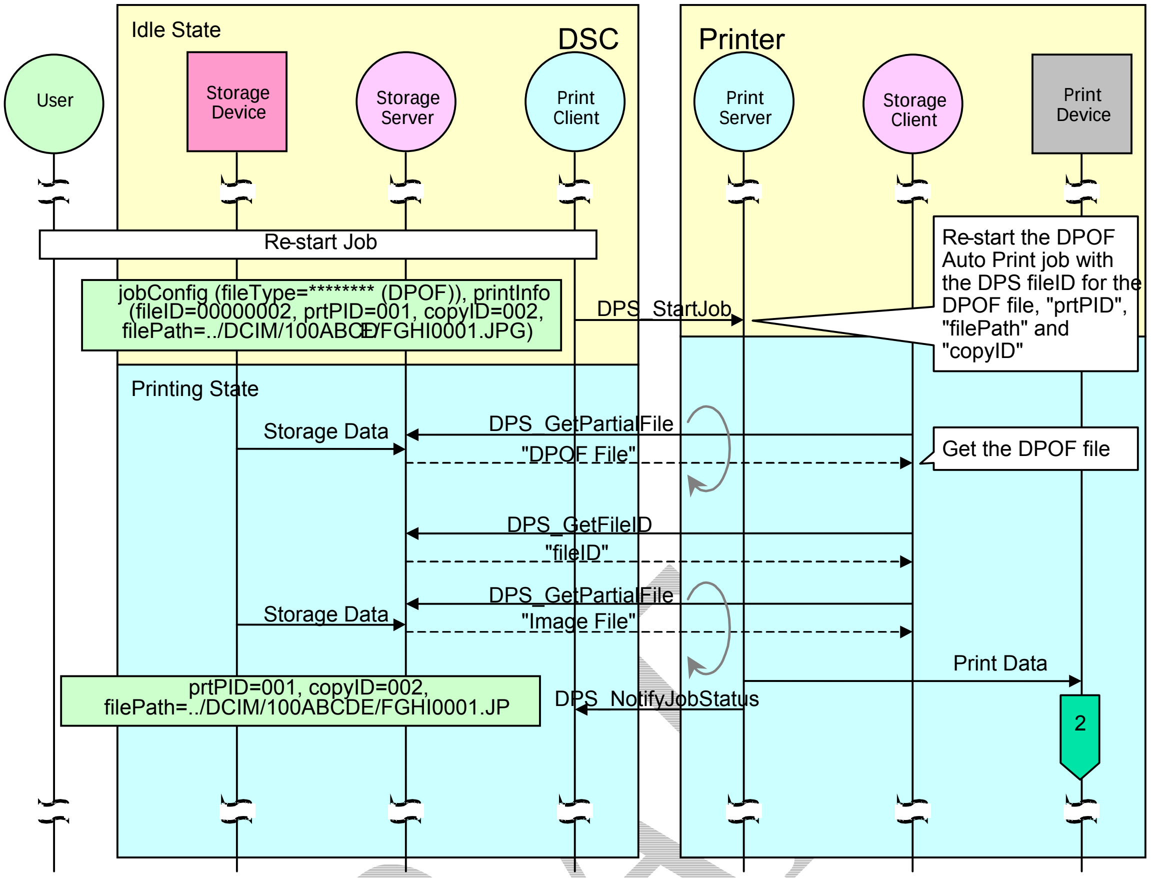
Idle State
DSC
Printer
User
Storage Device
Storage Server
Print Client
Print Server
Storage Client
Print Device
Re-start Job
jobConfig (fileType=******** (DPOF)), printInfo (fileID=00000002, prtPID=001, copyID=002, filePath=../DCIM/100ABCDE/FGHI0001.JPG)
DPS_StartJob
Re-start the DPOF Auto Print job with the DPS fileID for the DPOF file, "prtPID", "filePath" and "copyID"
Printing State
Storage Data
DPS_GetPartialFile
"DPOF File"
Get the DPOF file
DPS_GetFileID
"fileID"
Storage Data
DPS_GetPartialFile
"Image File"
Print Data
prtPID=001, copyID=002, filePath=../DCIM/100ABCDE/FGHI0001.JP
DPS_NotifyJobStatus
2

Figure C.1-4 Restarting a DPOF AutoPrint Job (Part 2)

Appendix D. Hardware Requirements (Normative)

The following hardware requirements apply to DPS Version 1.1:

D.1. Printer Requirements

1. USB host port – Standard A receptacle
2. Suspend/Standby
    - If DSC supports the remote wakeup capability, Printer can go to suspend mode
3. At least a 64K byte input buffer
4. Highly recommended to supply USB bus power: 5V, 500mA
5. Hub support is not required
6. Sufficient user interface to:
    - Indicate DPS connection
    - Clear paper errors and resume

D.2. DSC Requirements

1. USB device port – Standard Mini-B receptacle is highly recommended
2. At least a 1K byte input buffer if the DSC does not support XHTML-Print.
   At least a 4K byte input buffer if the DSC does support XHTML-Print.
3. Sufficient user interface to:
    - Indicate DPS connection
    - Indicate error has occurred

Appendix E. DPS System Features (Normative)

E.1. DPS Minimum System Features

This table describes Mandatory features of a DPS minimum system.

Feature	DPS Minimum System	Description	Note
Single Image Print	M	Print at least one image	If DSC has an image viewer, image currently viewed should be printed; else last image captured should be printed
DPS Connection Established	M	DPS system informs user connection established	
Error Occurred	M	DPS system informs user error occurred	
Paper-out recover trigger in printer	M	Printer enables user to recover from paper out condition	

Table E.1-1 DPS Minimum System Features

This table describes each feature is Mandatory/Recommended/Optional for Minimum DPS system.

	DPS System Features	DPS Minimum System	DPS Camera Support	DPS Printer Support
1. Basic Print				
1. Single Image Print	M	M	M	M
2. Several selected Images Print	R		R	M
3. DPOF AUTPRINT Images Print	R		R	R
4. All Images index print	R		R	R
5. All Images Print	R		R	M
6. XHTML-Print	O		O	O
2. Advanced print setup				
1. Cropping print	R		R	R
2. Copies	R		R	M
3. Date Print	R		R	R
4. Paper size	R		R	M
5. Fixed Size print	R		R	R
6. N-up layout	O		O	R
7. File name print	O		O	R
8. Borderless	R		O	R
3. Printer status information				
1. DPS connection established	M	M	M indicate	M indicate
2. Error occurred	M	M	M indicate	M indicate
3. Error Status	R		R indicate	M notify
4. Job status	R		R indicate	M notify
5. Cable disconnect enable	R		R indicate	R notify
6. Job Finish	R		R indicate	M notify
4. Printer setup				
1. Print quality	O		O	O
2. Paper type	O		O	O
5. Print job control				
1. DPS print cancel from DSC	R		R	M support
2. DPOF cable disconnect recovery	O		O	O
3. Paper Out Recover trigger in Printer	M	M	NA	M
4. Error recover UI in DSC	R		R	NA

Table E.1-2 Detailed DPS Minimum system features

In Table E.1-2, "notify" means that the Printer shall (if the entry is "M notify") or should (if the entry is "R notify") inform the DSC of the situation via the DPS protocol; "indicate" means the device shall (if the entry is "M indicate") or should (if the entry is "R indicate") provide the information to the user in some manner.

E.2. Print Result Compatibility

DPS does not define print result compatibility. However, a certain level of output that users can be satisfied with should be achieved. The following points may be vendor-dependent.

- Borderless or bordered, layout of images and the order of printing for the n-Up layout
- Border width for Bordered Printing
- Print area for the 1-Up Borderless Printing
- Position to be printed, font, size and color for date print and file name print

When the DSC needs text output, more control over layout, or a combination of text and images, XHTML-Print is recommended. For further information about controlling image layout and text output refer to Chapter 11 and Appendices A and B in [MIPC], *Implementation Guidelines for Printing with Mobile Terminals, Version 2.0*. DPS expects updated versions of these Guidelines to be published periodically by the Mobile Imaging and Printing Consortium, Inc. (http://www.mobileprinting.org/). Please check the Consortium's website for newer information that may be relevant to use of XHTML-Print.

Appendix F. Reference Table of Common Names for Fixed Sizes (Normative)

F.1.Common Names for Fixed Sizes

Paper Size	Common Name			Remark
	US	Japan	Europe	
2 ½" x 3 ¼"	2 ½ x 3 ¼			Wallet
3.5" x 5", 127 mm x 89 mm	3½ x 5	L	L	Standard Photo Size
4" x 6"	4 x 6			Photo
5" x 7", 178 mm x127 mm	5 x 7	2L	2L	Double-L Size
8" x 10"	8 x10			Photo
254 mm x178 mm		4L		Quad-L Size
110 mm x 74 mm		E		Standard Photo Size
89 mm x 55 mm		Card		Name Card Size
100 mmx148 mm		Hagaki Post Card		
6 cm x 8 cm			6 x 8 cm	
7 cm x 10 cm			7 x 10 cm	
9 cm x 13 cm			9 x 13 cm	
10 cm x15 cm			10 x 15 cm	
13 cm x18 cm			13 x 18 cm	
15 cm x 21 cm			15 x 21 cm	
18 cm x 24 cm			18 x 24 cm	
A4		A4	A4	
Letter	Letter			

Figure F.1-1 Common Names for Fixed Sizes

Appendix G. pictbridge URI scheme (Normative)

This section defines the URI (Uniform Resource Identifier) scheme “pictbridge”. For more details about URI, see [URI].

A “pictbridge” URI is used to identify resources in the DPS system. In other words, it is used to refer to files in the DPS Storage Server from an XHTML-Print file that has the URI.

G.1. pictbridge URI syntax

The pictbridge scheme uses the URI syntax defined in RFC 3986, Appendix A, “Collected ABNF for URI” with the following modifications. For more details about ABNF, see [ABNF].

```
scheme       = "pictbridge"                              (1)
authority    = [ userinfo "@" ] host [ ":" port ]        (2)
host         = reg-name                                  (3)
reg-name     = "localhost"                               (4)
             / "storageserver"
```

Description of the preceding productions relative to RFC 3986:

(1) Declares the scheme name.

(2) The pictbridge scheme assigns no meaning to the optional userinfo subcomponent in ‘userinfo "@"‘. The optional port subcomponent in ‘ ":" port ‘ has no meaning in the DPS/PTP context, and therefore shall be ignored by the XHTML parser (but not treated as invalid).

(3) The pictbridge scheme does not recognize either an IPv4 or IPv6 address as a valid host.

(4) The pictbridge scheme recognizes only the two strings “localhost” and “storageserver” as a valid reg-name for a host. The reg-name “localhost” means the same as it does in the file and http schemes, it refers to the device processing the XHTML file. When used in the context of DPS 1.1, this device will be the printer, but the pictbridge scheme makes no assumption about and imposes no requirements on the characteristics of the processing device. The reg-name “storageserver” is created from the system component name “Storage Server” in section 5.1. It is not capitalized and not hyphenated to conform to the structure of the name “localhost”.

The path component represents the logical hierarchical structure in the DPS Storage Server. The Storage Server can freely design where the root directory is in the path component.

e.g., “/DCIM/100ABCDE/ FGHI0001.JPG”

e.g., “/00010001/DCIM/100ABCDE/ FGHI0001.JPG”

Special processing is applied when the string “vnd.osta-org.exif-thumb” (See [MPVCORE]), appears as the fragment component of a URI. This fragment indicates that the URI references the thumbnail of an EXIF file (hereinafter called “exif-thumb”). This special processing is described in section G.2.1.

G.2. Dereference

G.2.1. Dereference of pictbridge scheme for “storageserver” host

The steps to dereference a pictbridge-based URI are as follows.

(1) Before processing any URIs in the XHTML-Print document, establish the base URI according to RFC 3986, Section 5.1. The retrieval URI mentioned in Section 5.1.3 will be the URI provided as the value of filePath in DPS_StartJob.

(2) For each relative URI in the XHTML-Print document, convert it to an absolute URI according the procedures in the rest of RFC 3986, Section 5.

(3) The `filePath` parameter in DPS_GetFileID is set to the absolute URI without the fragment component.

Then the DPS_GetFileID operation is executed.

(4) The `fileID` parameter in following operations is set to the `fileID` parameter in the response of the DPS_GetFileID. Then following operations are executed.

- DPS_GetThumb (when an exif-thumb fragment is present)
- DPS_GetFile/DPS_GetPartialFile (when an exif-thumb fragment is not present)

G.3. pictbridge URI examples

G.3.1. example of pictbridge scheme for “storageserver” host

Common example:

pictbridge://storageserver/DCIM/100ABCDE/ FGHI0001.JPG

A thumbnail specified example:

pictbridge://storageserver/FGHI0001.JPG#vnd.osta-org.exif-thumb

Appendix H. DPS Support for XHTML-Print (Normative)

This section describes the normative information regarding implementation of XHTML-Print in DPS. See [XHTML-Print] for more information. In order to support XHTML-Print in DPS, it is necessary to be capable of handling URI. However, the Printer cannot directly get the object of each referenced file using its URI because DPS adopts a handle-based system to specify a particular file object. Consequently, this specification defines a way of converting from a particular URI to its DPS fileID.

The DPS does not support multiplexed inline image data as described in RFC3391 - The MIME Application/Vnd.pwg-multiplexed Content-type, see [MIMEMPX].

H.1. Implementation of processing the XHTML-Print file in DPS

H.1.1. DPS Operations Used

The following operations are required to process XHTML-Print jobs. The DPS system defines an operation, DPS_GetFileID, to convert a URI of a file as provided in the `filePath` in the DPS_StartJob and the XHTML-Print file, to a DPS fileID, on the occasion when using an XHTML-Print parser in the Printer. See section 8.2.1 for more information about DPS_GetFileID.

- DPS_ConfigurePrintService
- DPS_GetCapability
- DPS_StartJob
- DPS_GetPartialFile/DPS_GetFile
- DPS_GetFileID
- DPS_GetDeviceStatus/DPS_NotifyDeviceStatus
- DPS_GetJobStatus/DPS_NotifyJobStatus

H.1.2. Processing sequence for XHTML-Print

H.1.2.1. XHTML-Print capability confirmation

Before starting XHTML-Print processing, the DSC should determine whether or not the Printer supports XHTML-Print by getting the list of `fileTypes` parameters in DPS_GetCapability response from the Printer. If the Printer supports XHTML-Print, it returns `fileTypes` parameters that include "****xxxx" (XHTML-Print). The fileTypes minor code (xxxx) indicates the character encodings the Printer supports. For more details about this, see section10.5.2. Then if necessary, the DSC can determine printable character repertoire the Printer supports. See section 10.5.3. The DSC also makes sure whether the Printer supports `fileTypes` referenced in the XHTML-Print file.

An execution example is shown below:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getCapability>
   <capability>
    <fileTypes/>
   </capability>
  </getCapability>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getCapability>
   <capability>
    <fileTypes> ******** ******** ******** ********</fileTypes>
<!-- fileTypes: ****xxxx (XHTML-Print)
     Default(0011****), UTF-8(****) and Shift_JIS(****) support) -->
   </capability>
  </getCapability>
 </output>
</dps>
```

H.1.2.2. Primary process

This section describes how an XHTML-Print job is processed in the DPS system. The following is an example showing a typical file system on a DSC that complies with DCF and a typical XHTML-Print file. The sample DPS scripts in this explanation assume the following file structure and XHTML-Print file.

Note: The XHTML-Print document SAMPLE.HTM is not intended to produce printed output that is acceptable to end users. How to create a quality XHTML-Print document is beyond the scope of this specification. Refer to [MIPC] for that information.

```
/SAMPLE.HTM
/DCIM/100ABCDE/FGHI0001.JPG
/DCIM/100ABCDE/FGHI0002.JPG
```

Figure H.1-1 DCF File System Directory Example

DPS fileID	File or Folder Description
00000002	/SAMPLE.HTM
00000005	/DCIM/100ABCDE/FGHI0001.JPG
00000006	/DCIM/100ABCDE/FGHI0002.JPG

Table H.1-1 DCF File System Example

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML-Print 1.0//EN"
    "http://www.w3.org/MarkUp/DTD/xhtml-print10.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">

<head>
<title>Sample</title>
</head>
<body>
<img src="DCIM/100ABCDE/FGHI0001.JPG" alt="image 1"/>
<img src="DCIM/100ABCDE/FGHI0002.JPG" alt="image 2"/>
</body>
</html>
```

Figure H.1-2 SAMPLE.HTM

1) Start an XHTML-Print Job

The DSC starts an XHTML-Print job by issuing the DPS_StartJob operation. The `fileID` parameter shall be set to ********, which indicates that the `filePath` parameter contains the absolute URI for the XHTML-Print file.

An execution example is shown below:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
 <startJob>
   <jobConfig>
    <fileType>********</fileType>
<!-- fileType:  ****xxxx (XHTML-Print) Default(****) -->
   </jobConfig>
   <printInfo>
    <fileID>********</fileID>
    <filePath>pictbridge://storageserver/SAMPLE.HTM</filePath>
   </printInfo>
  </startJob>
 </input>
</dps>
```

2) Get a DPS fileID for the XHTML-Print file

When the Printer encounters the `fileType` parameter in DPS_StartJob that is XHTML-Print, the Printer shall confirm that the `filePath` parameter in printInfo is an absolute URI. When the URI contains the pictbridge scheme, it requires an operation to ask the DSC to convert the URI into a valid DPS fileID. The operation is DPS_GetFileID. When the `filePath` in DPS_GetFileID is set to the URI provided in `filePath` in DPS_StartJob, the `basePathID` parameter in DPS_GetFileID is necessary to be set to ********. The Printer calls it to get a usable DPS fileID from the URI. This URI is retrieval URI described in Section 5.1.3 of RFC 3986 that is used to establish the base URI for relative URIs.

Implementer's Note:

- As described in RFC 3986, the scheme component and host subcomponent of a URI are case-insensitive. When reading a URI, devices should treat each upper case character in the scheme and host as its lower case equivalent. When writing a URI, all devices should use only lower case characters in scheme and host, except for percent-encoded characters. The hexadecimal characters ‘a’ thru ‘f’ should be written in their upper case form as ‘A’ thru ‘F’ in percent-encodings.
- All other components and subcomponents are case-sensitive. Their input case shall be preserved on output.

An execution example is shown below:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <input>
  <getFileID>
   <basePathID>********</basePathID>
   <filePath>pictbridge://storageserver/SAMPLE.HTM</filePath>
  </getFileID>
 </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getFileID>
   <fileID>00000002</fileID>
  </getFileID>
 </output>
</dps>
```

The `fileID` parameter value of 00000002 is the DPS fileID of the URI requested by the Printer.

3) Get the XHTML-Print file

The Printer then retrieves the XHTML-Print file using DPS_GetFile or DPS_GetPartialFile for the DPS fileID returned by DPS_GetFileID.

4) Get a DPS fileID for a referenced file

Any URI in the XHTML-Print file that uses the pictbridge scheme and references an object that the Printer must fetch (such as an image or a stylesheet), it requires the DPS_GetFileID operation to ask the DSC to convert the URI into a valid DPS fileID. The Printer calls it to get a usable DPS fileID from the URI. When the `filePath` is set to the absolute URI that contains the pictbridge scheme, the `basePathID` parameter in DPS_GetFileID is necessary to be set to ********.

Implementer's Note:

- If a fragment component is contained in the URI, the printer removes it before setting `filePath` in DPS_GetFileID to that URI.
- If the URI refers to a stylesheet, or any other object that might itself contain relative URIs, the printer shall use that object’s absolute URI as the retrieval URI for establishing the base URI for all relative URIs contained therein.

An execution example is shown below:

Input Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
```

```
  <input>
   <getFileID>
    <basePathID>********</basePathID>
    <filePath>pictbridge://storageserver/DCIM/100ABCDE/FGHI0001.JPG
</filePath>
   </getFileID>
  </input>
</dps>
```

Output Example:

```
<?xml version="1.0"?>
<dps xmlns="http://www.cipa.jp/dps/schema/">
 <output>
  <result>********</result>
  <getFileID>
   <fileID>00000005</fileID>
  </getFileID>
 </output>
</dps>
```

The `fileID` parameter value of 00000005 is the DPS fileID of the URI requested by the Printer.

5) Get a referenced file

The Printer then retrieves the referenced file for printing using DPS_GetFile or DPS_GetPartialFile for the DPS fileID returned by DPS_GetFileID.

6) Repeat 4)-5) if multiple image files to be printed are referenced in the XHTML-Print file.

7) Print the referenced files according to the layout specified in XHTML-Print File.

Figure H.1-3 shows the state transition and sequence diagram for processing an XHTML-Print job.

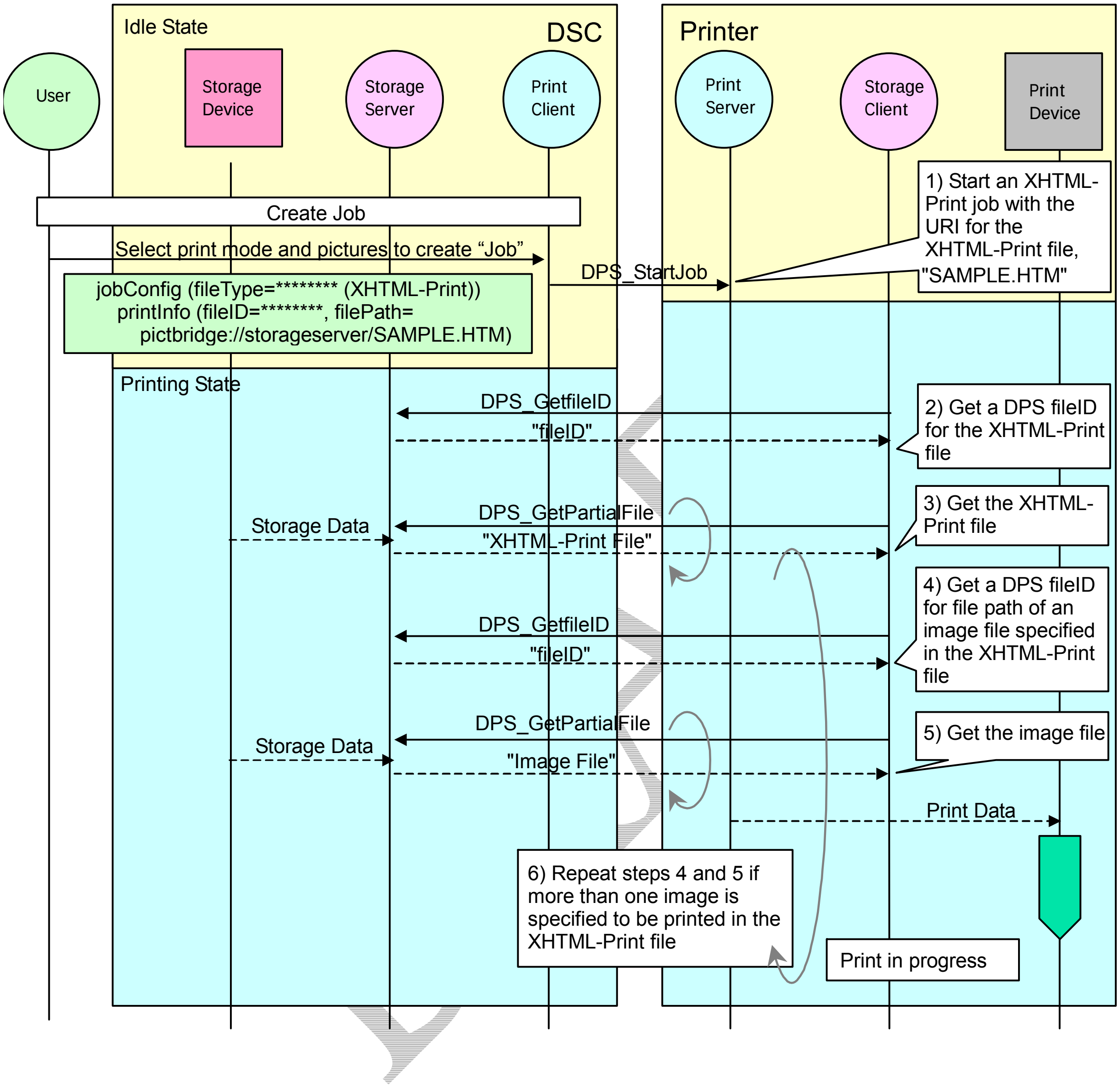
Idle State
DSC
User
Storage Device
Storage Server
Print Client
Printer
Print Server
Storage Client
Print Device
Create Job
Select print mode and pictures to create "Job"
jobConfig (fileType=******** (XHTML-Print))
printInfo (fileID=********, filePath=
pictbridge://storageserver/SAMPLE.HTM)
DPS_StartJob
1) Start an XHTML-Print job with the URI for the XHTML-Print file, "SAMPLE.HTM"
Printing State
DPS_GetfileID
"fileID"
2) Get a DPS fileID for the XHTML-Print file
DPS_GetPartialFile
Storage Data
"XHTML-Print File"
3) Get the XHTML-Print file
DPS_GetfileID
"fileID"
4) Get a DPS fileID for file path of an image file specified in the XHTML-Print file
DPS_GetPartialFile
Storage Data
"Image File"
5) Get the image file
Print Data
6) Repeat steps 4 and 5 if more than one image is specified to be printed in the XHTML-Print file
Print in progress

Figure H.1-3 Print Transitions on XHTML-Print Processing